PHP+MySQL 动态网站开发与全程实例

聂庆鹏 毛书朋 王志乐 编著

- 以最佳的学习方式，讲解PHP最常用、最实用的功能
- 功能讲解与实际项目结合，用一个大型案例串讲全书各章示例
- 讲解开发思路、技术实现过程，教你掌握贴近技术前沿的知识和技能
- 113个课堂演示实例，9个 上机实验，1个网络应用综合实例

清华大学出版社

PHP+MySQL
动态网站开发与全程实例

聂庆鹏 毛书朋 王志乐 编著

清华大学出版社
北 京

内 容 简 介

本书来自作者多年学习、应用和讲授 PHP 的经验与体会，是专为学习 PHP＋MySQL 数据库编程人员编写的入门教材。

全书共分 10 章，循序渐进地介绍 HTML 基础、Javascritp 基础、PHP 安装和服务器环境配置（包括 IIS 和 Apache）、PHP 语法、内置数组、常用函数、文件目录操作，MySQL 数据库安装使用、PHP＋MySQL 数据库编程等内容；在最后二章设计了 2 个贴近实际应用的典型案例：留言本系统和论坛系统，每个案例先介绍开发思路、步骤，再给出全部源代码，使所学内容与实际应用紧密结合，特别是论坛系统将全书的案例串讲起来，力求使读者学到最贴近应用前沿的知识和技能。

本书在内容选择上，不求面广，但求实用；在讲述方式上，突出案例教学，避免空洞的描述，每章都根据章节内容设计一个典型案例和若干演示案例，并留有帮读者梳理所学知识的练习题、上机题，力求通过课堂演练和上机练习，全面提高学生的学习效果和动手能力。

本书可用作高等院校相关课程的教学用书和教学辅导书，也是 PHP 动态网站开发的最佳入门教材。

图书在版编目（CIP）数据

PHP+MySQL 动态网站开发与全程实例/聂庆鹏等编著. —北京：清华大学出版社，2007

ISBN 978-7-302-16213-1

Ⅰ. P…　Ⅱ. 聂…　Ⅲ. ①PHP 语言-程序设计　②关系数据库-数据库管理系统，MySQL　Ⅳ. TP312　TP311.138

中国版本图书馆 CIP 数据核字（2007）第 152889 号

责任编辑：夏非彼　范　蕾
责任校对：贾淑媛
责任印制：
出版发行：清华大学出版社　　**地　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编：**100084
c – service@tup.tsinghua.edu.cn
社 总 机：010-62770175　　**邮购热线：**010-62786544
投稿咨询：010-62772015　　**客户服务：**010-62776969
印 装 者：
经　销：全国新华书店
开　本：185×260　**印　张：**21　**字　数：**510 千字
版　次：2007 年 10 月第 1 版　**印　次：**2007 年 10 月第 1 次印刷
印　数：1~4 000
定　价：29.80 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：（010）62770177 转 3103　产品编号：027310-01

前　言

PHP是目前全世界最为流行的Web开发语言之一。最近几年，PHP在国内的发展更是十分迅速，国内几乎所有的虚拟主机和相当数量的服务器都提供了PHP支持。PHP作为一门功能强大的Web编程语言，以其简单易学、安全性高以及跨平台等诸多特性而越来越受到广大Web开发者的关注和喜爱。

现在，越来越多的人开始关注PHP、学习PHP、使用PHP。但是相对于其他Web开发语言，专门介绍PHP的书籍却一直偏少，很多PHP初学者都苦于找不到一本通俗易懂、简单实用的入门教材。笔者近五六年时间都在学习、使用、讲授PHP，积累了比较丰富的经验，因此，编写一本PHP教程来引导初学者快速入门，就觉得比较有意义。因此我们组织多名有丰富开发经验的人员共同拟定了这本书的写作大纲，筛选适于教学的开发案例，希望这本书可以帮助那些喜爱PHP的朋友快速走上学习PHP的捷径。

本书共分10章，循序渐进地介绍了PHP概况、HTML基础知识、PHP的安装配置、PHP语法、内置数组与常量、PHP函数库、文件系统操作、MySQL数据库安装与使用、PHP+MySQL数据库编程等内容。本书最后一章设计了一个简单论坛系统的实例，详细介绍其开发步骤，同时也将前面各章的主要实例进行串讲，并给出了全部源代码，力求使读者能够学到最贴近应用前沿的知识和技能。

本书总结了编者近几年来学习和使用PHP的经验心得。内容选取上以实用性为原则，不求面广，但求实用。本书突出案例教学，避免空洞的描述，每章都根据章节内容设计一个典型案例，通过对案例的深入分析和上机操作使读者加深对所学知识的理解，提高学习效果和动手能力。

本书定位为高等院校计算机类学生的专业课教材，也可以作为各类希望学习Web开发技术人员的自学教材，同时还可以作为初、中级PHP开发者的参考书。

本书由聂庆鹏、毛书朋主编，郑国华、王志乐任副主编。本书第3章、第6章、第9章由聂庆鹏编写；第4章、第5章、第10章由毛书朋编写；第1章、第2章由王志乐编写；第7章、第8章由郑国华编写。全书最后由聂庆鹏汇总定稿。

由于作者水平有限，书中难免存在疏漏和错误之处，恳请专家和广大读者批评指正。在学习过程中，遇到疑难问题，可以通过以下方式与我们联系：booksaga@126.com，也可以登录图格网站http://www.booksaga.com留言，我们将在第一时间给予答复！

为方便教学，聂庆鹏老师亲自制作了与本书配套使用的PPT教学课件，有需求的老师可通过电子邮件与我们联系。

编　者

2007年3月

目　　录

CHAPTER 1

PHP 概述



本章重点

- PHP 的特点及其在动态网站开发中的应用
- PHP 常用开发工具

1.1 PHP 的来龙去脉

PHP 到底是怎样的一种语言，有怎样的产生和发展过程呢？让我们带着这两个问题来开始我们的学习吧。

1.1.1 PHP 的概念

PHP 是 Hypertext Preprocessor（超文本预处理器）的字母缩写，是一种跨平台的开放源代码的服务器端嵌入式脚本语言。它借鉴了 C、Java、Perl 等语言的部分语法并结合 PHP 自己的特性，使 Web 开发者能够快速地写出动态生成页面的脚本。

与 HTML 不同的是，PHP 代码是运行在服务器端的，然后把结果返还到客户端来。使用 PHP 的一大好处是它对于初学者来说极其简单，可以很快的入门，只需几个小时就可以自己写一些简单的脚本，同时也给专业的程序员提供了各种高级的特性。

尽管 PHP 的开发是以服务器端脚本为目的，但事实上其功能远不局限与此。请继续读后面的章节，在 1.2 节“PHP 可以做哪些事”一节中将获得更多的信息。

1.1.2 PHP 的产生与发展

任何事物都有其产生和发展的过程，PHP 也不例外。但 PHP 与其他脚本语言的不同在于，PHP 能够诞生并且发展到今天的规模，是全世界各位免费代码开发者共同努力的结果。它是通过 Internet 把众多程序员的编程精粹汇聚，再由 PHP 开发者将其转化到程序设计语言中才诞生的，可以说，PHP 是 Internet 的产物，是开放源代码自由交流的胜利成果。

PHP 的创建者是 Rasmus Lerdorf，一开始它只是一个由 Perl 语言编写的小程序，用于计算网页访问量。后来又用 C 语言重新编写，并扩展了数据库功能。随着 PHP 使用范围的不断扩大，在 1995 年，Rasmus Lerdorf 开始对外发布第一个版本。在此版本中，提供了访客留言本、访客计数器等简单的功能。

继 PHP 第一个版本问世后，一些程序员也开始加入了 PHP 的代码设计，随后第二版的 PHP 问世，定名为 PHP/FI。在此版本中，加入了对数据库 MySQL 的支持，自此奠定了 PHP 开发动态网页的基础。到 1996 年底，已经有 15000 多个网站使用 PHP/FI，到 1997 年，数量达到 50000 个。

PHP 3 的问世使 PHP 有了快速的发展，它是 PHP 的第三个版本，诞生于 1997 年，在 Zeev Suraski 及 Andi Gutmans 等人加入到 PHP 开发小组之后，对 PHP 进行了彻底崭新的设计，加入了众多的新功能和新技术而推出的。PHP 3 能够与 Apache 服务器紧密结合，并且支持几乎所有的主流与非主流数据库。另外，它的执行效率大大提高，1999 年 PHP 在网站中的应用数量超过了 150000 个。

PHP 4 的推出使 PHP 更加的成熟，整个脚本程序的核心大幅度改动，使用了一种“编译—执行”模式，核心引擎更加优越，执行速度更快，已比传统的 CGI 及 ASP 等程序有更好的表现，而且有更新的功能、更丰富的函数库。

PHP 进入中国时间较晚，在相当长的时间内，PHP 在国内使用率很低。但是最近几年，

PHP 以其易学、高效、安全、免费、跨平台等一系列重要优势，在国内迅速发展开来，现在浏览国内网站不难发现，PHP 的踪迹无处不在，很多大的网站都是用 PHP 作为开发语言。

PHP 目前的最新版本是 PHP 5，它在 PHP 4 的基础上进行了进一步的改进，功能更强大，执行效率更高。如果您想获得更多更新的 PHP 相关资料，可以到 PHP 官方网站 http://www.php.net 上浏览，相信肯定大有收获。

1.2　PHP 可以做哪些事

通过上面的学习中我们对 PHP 有了一个大致的了解，这一节就来学习 PHP 到底可以做哪些事。

如果说 PHP 无所不能，大概是夸大了些，但如果说在 Web 开发上 PHP 能完成任何一款服务器端程序所能完成的工作却一点也不为过，例如，收集表单数据，生成动态网页，或者发送/接受 Cookies。当然，PHP 的功能不只局限于此。

PHP 脚本主要用于以下三个领域。

- **服务端脚本**：这是 PHP 最传统，也是最主要的目标领域。开展这项工作需要具备以下三项配置：PHP 解析器（CGI 或者服务器模块）、Web 服务器和 Web 浏览器。需要在运行 Web 服务器时，安装并配置 PHP，然后，可以用 Web 浏览器来访问 PHP 程序的输出，即浏览您服务端的 PHP 页面。如果只是实验 PHP 编程，所有的这些都可以运行在自己家里的电脑中。
- **命令行脚本**：可以编写一段 PHP 脚本，并且不需要任何服务器或者浏览器来运行它，仅仅只需要 PHP 解析器来执行。这种用法对于依赖 Cron（Unix 或者 Linux 环境）或者 Task Scheduler（Windows 环境）的日常运行的脚本来说是理想的选择。这些脚本也可以用来处理简单的文本。
- **编写桌面应用程序**：对于有着图形界面的桌面应用程序来说，PHP 或许不是一种最好的语言，但是如果用户非常精通 PHP，并且希望在客户端应用程序中使用 PHP 的一些高级特性，则可以利用 PHP-GTK 来编写这些程序。用这种方法，还可以编写跨平台的应用程序。PHP-GTK 是 PHP 的一个扩展，在通常发布的 PHP 包中并不包含它。如果对 PHP-GTK 感兴趣，请访问其网站以获取更多信息。

有了 PHP，我们可以轻松地进行 Web 开发，轻松地制作动态网站、新闻系统、留言板、论坛 BBS、聊天室、博客等；轻松地实现对文件及文件夹的操作；轻松地进行电子商务的开发、数据的加密；与数据库相关联，处理大批量的数据等等。我们将在以后的章节中向广大读者介绍 PHP 这些激动人心的功能。

1.3　PHP 的特点及与其他语言的比较

1.3.1　PHP 的特点

前面两节我们学习了 PHP 的概念及功能，这一节我们重点学习 PHP 的特点。PHP 自产

生以来一直都在发展中应用，在应用中发展，这是因为 PHP 不仅有着其他同类脚本所共有的功能，更有它自身的特色。它的特色如下。

- **完全免费**：使用 PHP 进行 Web 开发无需支付任何的费用。
- **代码完全开放**：所有的 PHP 程序代码都可以免费地自由地交流。在 Internet 上，你可以得到更多的 PHP 源程序，为自己所用。
- **语法结构简单**：PHP 大量结合了 C 语言和 Perl 语言的特色，坚持以基础语言开发程序，编写方便易懂。对于以前接触过 C 语言的用户来说，只需要了解 PHP 的基本语法，再掌握些 PHP 独有的函数，就可以轻松踏上 PHP 程序设计之旅。
- **功能强大**：在 1.2 节的“PHP 可以做哪些事”里面已经介绍过，这里不再重复。
- **强大的数据库支持**：支持几乎所有的主流与非主流数据库，使 PHP 最大程度地被不同方面的用户使用。
- **代码执行效率高**：与其他同类 CGI 比较，PHP 消耗更少的系统资源，尤其当 PHP 作为 Apache 服务器的内嵌模块运行时，服务器除了承担脚本解释负荷外，无须承担其他额外操作。

PHP 的先进之处还体现在以下几个方面。

- **学习过程短**：与 JSP 和 Perl 不同，你不必把头埋进 100 多页的文档中努力学习才可以写出一个象样的程序。只要了解一些基本的语法和语言特色，你就可以开始你的 PHP 编码之旅了。之后你在编码过程中如果遇到了什么麻烦，还可以再去翻阅相关文档。你只需要 30 分钟就可以将 PHP 的核心语言特点全部掌握，你可能已经非常了解 HTML，甚至你已经知道怎样用编辑设计软件或者手工来制作好看的 Web 站点。由于 PHP 代码能够无障碍的添加进你的站点，在你设计和维护站点的同时，你可以很轻松地加入 PHP 使得你的站点更加具有动态特性。
- **面向对象编程**：PHP 提供了类和对象，基于 Web 的编程工作非常需要面向对象编程能力。PHP 支持构造器、提取类等。
- **可伸缩性**：传统上网页的交互作用是通过 CGI 来实现的。CGI 程序的伸缩性不很理想，因为它为每一个正在运行的 CGI 程序开一个独立进程。解决方法就是将经常用来编写 CGI 程序的语言解释器编译进你的 Web 服务器（比如 mod_perl，JSP)。PHP 就可以以这种方式安装，虽然很少有人愿意这样以 CGI 方式安装它。内嵌的 PHP 可以具有更高的可伸缩性。

1.3.2 PHP 与其他 CGI 语言的比较

接下来，我们就把 PHP 与其他 CGI 语言在几个方面的比较情况以表格的形式展示出来，使得各位读者有直观的比较和认识。见表 1-1，此表格的数据并不是笔者通过实测总结出来的，实际上要对一种语言的各种性能进行很具体的实测是很复杂且很困难的，但是就笔者的使用经历，结合其他资料中的观点，还是基本能够得出大体的结论。虽然不一定极为精确，但还足以反映出各种语言的特点。

表 1-1　PHP 与 ASP/CGI/JSP/ISAPI 性能比较表

语言 性能指标	PHP	ASP	CGI	JSP	ISAPI
操作系统	均可	Win32	均可	均可	Win32
Web 服务器	多种	IIS	均可	数种	IIS
执行效率	快	快	慢	极快	极快
稳定性	佳	中等	最高	佳	差
开发时间	短	短	中等	较长	长
程序语言	PHP	VB	不限	Java	C/ Delphi
网页结合	佳	佳	差	差	差
学习门槛	低	低	高	高	高
函数支持	多	少	不定	多	少
系统安全	佳	差	佳	佳	尚可
升级速度	快	慢	无	较慢	慢

1.4　PHP 常用开发工具

编写 PHP 代码的工具很多，常用的网页编辑器如 FrontPage、Dreamweaver，常用的文本编辑器如 Word、UltraEdit，甚至用 Windows 自带的记事本都可以书写源代码，当然，还有专门的 PHP 开发工具，如 PHPEdit、PHPed、Dev-PHP 等，下面我们就介绍三种不同类型的 PHP 开发工具。

1.4.1　PHP 代码开发工具

1. EditPlus

速度快，支持多种语言的语法高亮，有 Project 概念，但并不好用。无 SCC（Source Code Control，源代码控制）集成，但仍是个优秀的编辑器，界面如图 1-1 所示。

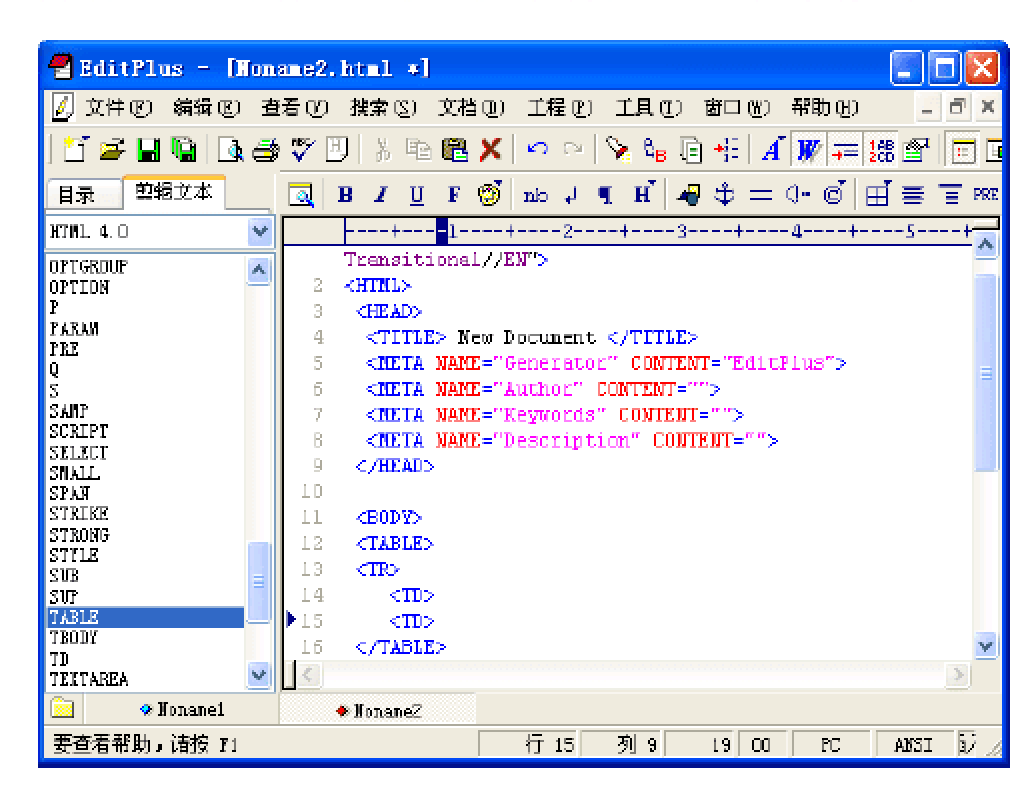


图 1-1　EditPlus 工具界面

2. Dev-PHP

用 Delphi 开发的 Opensource 的 PHP 开发工具，较好地集成了 PHP 解析器和 PHP-GTK 库，性能和稳定性都很不错，只是在团队合作上比较薄弱，没有 SCC 和 Project 的功能，但完全让你有理由舍弃掉 EditPlus。

3. PHPEdit

PHPEdit 是一款 Windows 下优秀的 PHP 脚本 IDE（集成开发环境）。该软件为快速、便捷地开发 PHP 脚本提供了多种工具，其功能包括：语法关键词高亮；代码提示、浏览；集成 PHP 调试工具；帮助生成器；自定义快捷方式；150 多个脚本命令；键盘模板；报告生成器；快速标记；插件等。现在最新的是 2.6.1 版本，安装包大小 30M 左右，安装时候只需要到其官方网站上免费注册一下就可以获得一个月的试用期。界面如图 1-2 所示。

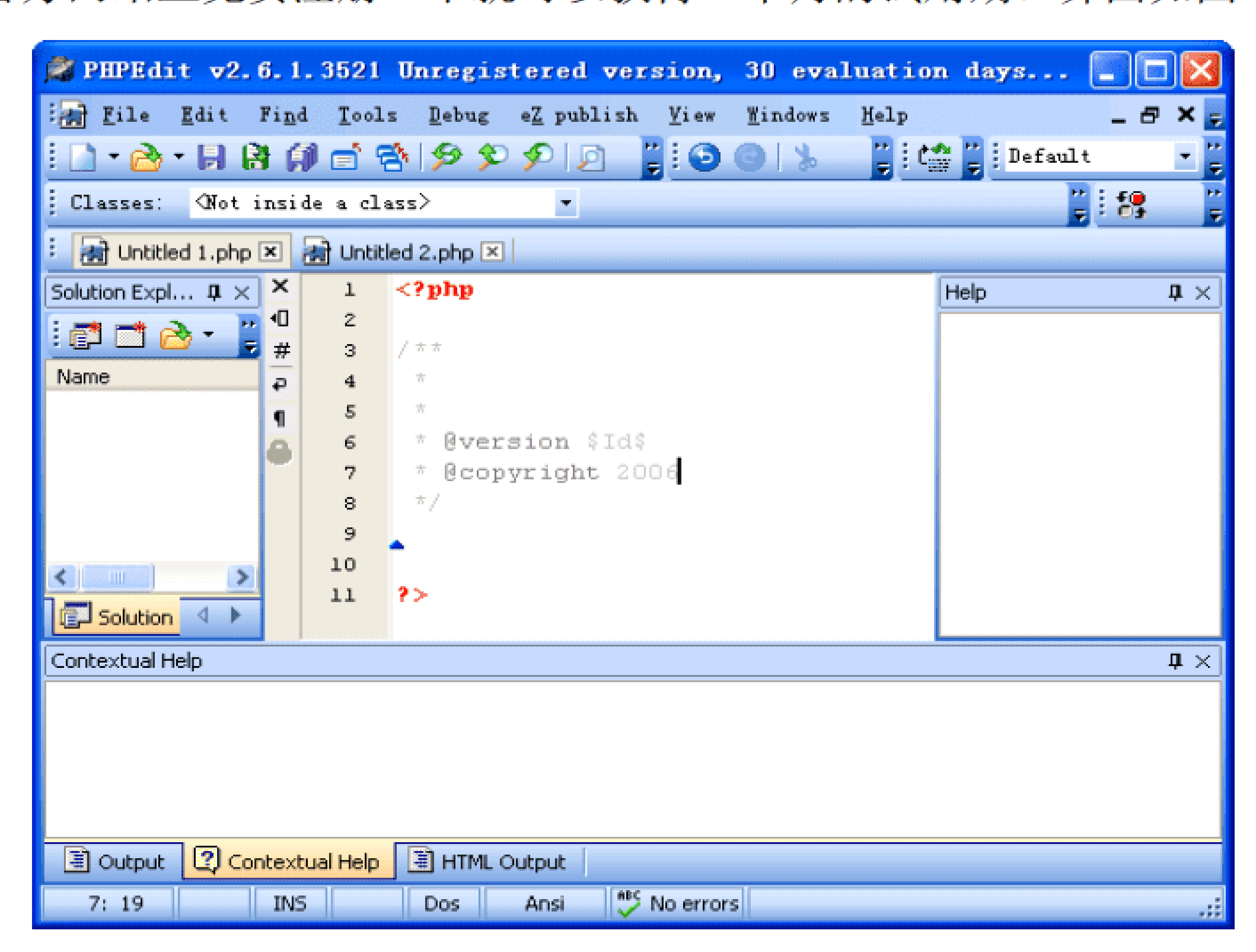


图 1-2 PHP Edit 工具界面

4. Zend Studio

一款优秀的 IDE。各个功能丰富，Zend 的拳头产品，带有 CVS 集成，项目管理也较为丰富，但作为商业软件，价格颇高，一般个人用户没有太大必要去购买。

1.4.2 网页设计工具

下面介绍两种简单的网页设计工具。

1. Microsoft FrontPage 2003

FrontPage 是 Microsoft 公司推出的 Office 系列中的一款制作网页的软件，简单易学，容易上手，有 Word 操作经验的人学起来都会觉得很容易。对于以前用 FrontPage 开发网页过程中出现的垃圾代码，相信用过的读者肯定深有体会。但 FrontPage 2003 的出现使这一问题得到了很大的改善。比起以前的版本，FrontPage 2003 的功能更强大，界面更友好（见图 1-3），产生的垃圾代码更少，开发效率更高。

2. Dreamweaver

作为网页三剑客之一的Dreamweaver，在网站的设计与开发上与FrontPage各有优缺点，界面和 FrontPage 虽有差别，但若对其中一个熟悉后，对另一个的学习就会事半功倍，这里也就不再赘述。

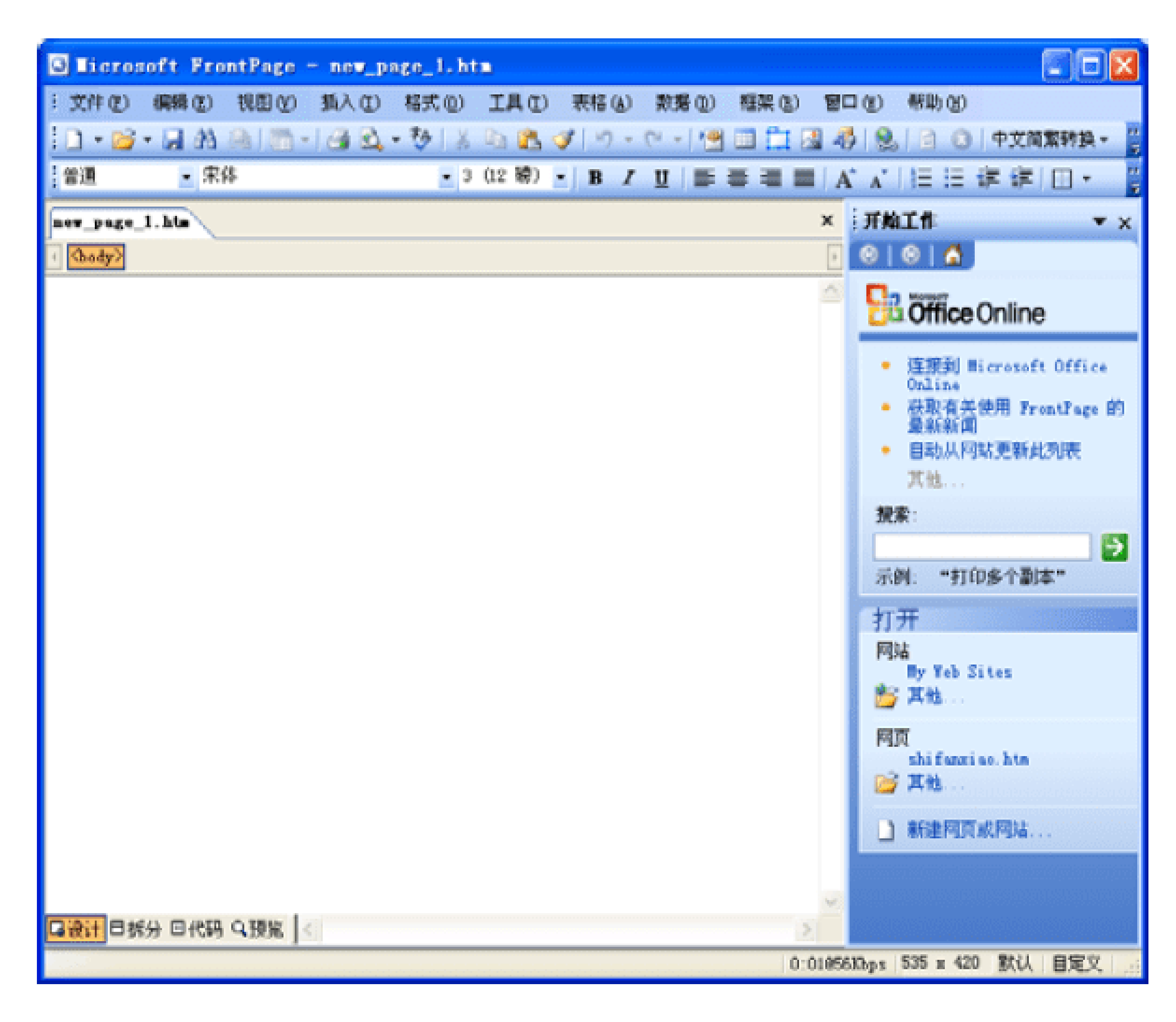


图 1-3　FrontPage 界面

1.4.3　文本编辑工具

1. UltraEdit

这款功能强大的文本编辑器可以编辑文字、Hex、ASCII 码，可以取代记事本，可同时编辑多个文件，而且即使开启很大的文件，速度也不会慢。最新版本的软件修正了老版本存在的一些 Bug，并新增了二十余项新功能，界面如图 1-4 所示。

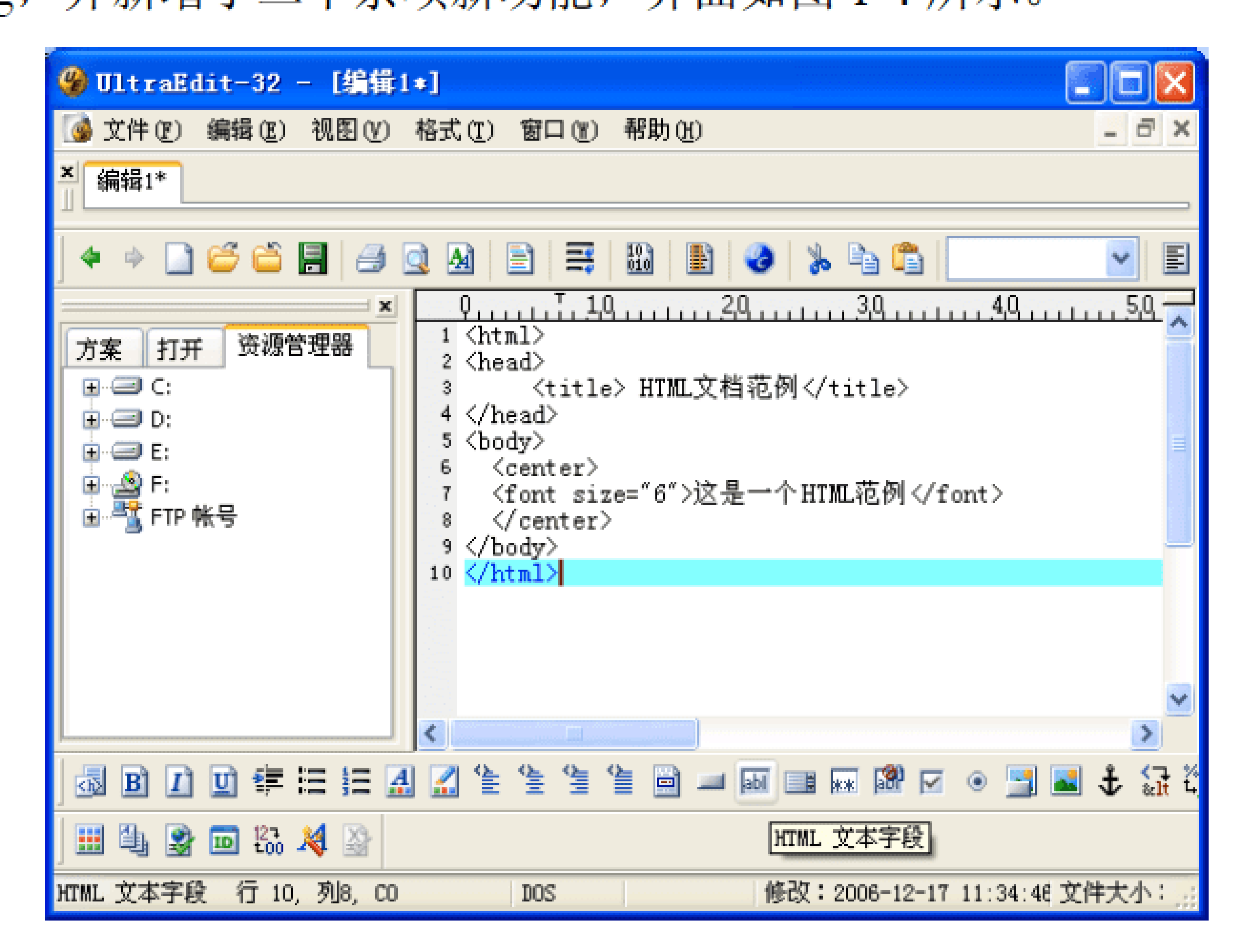


图 1-4　UltraEdit 工具界面

2. NotePad（记事本）

这是 Windows 自带的小软件，笔者通常在需要对源代码稍作修改的时候会用到它。记事本占用内存极少，运行速度极快，但功能较简单。当然，随着计算机硬件配备的不断提高和其他专用软件的兴起，它已慢慢淡出了我们的视线。在使用记事本编写好程序存盘的时候应注意，在文件名一栏手动输入文件名字，后缀为.php，保存类型应选择“所有文件”（见图 1-5），否则系统会自动为文件加上后缀名.txt，使之成为文本文件，无法实现 PHP 的功能。

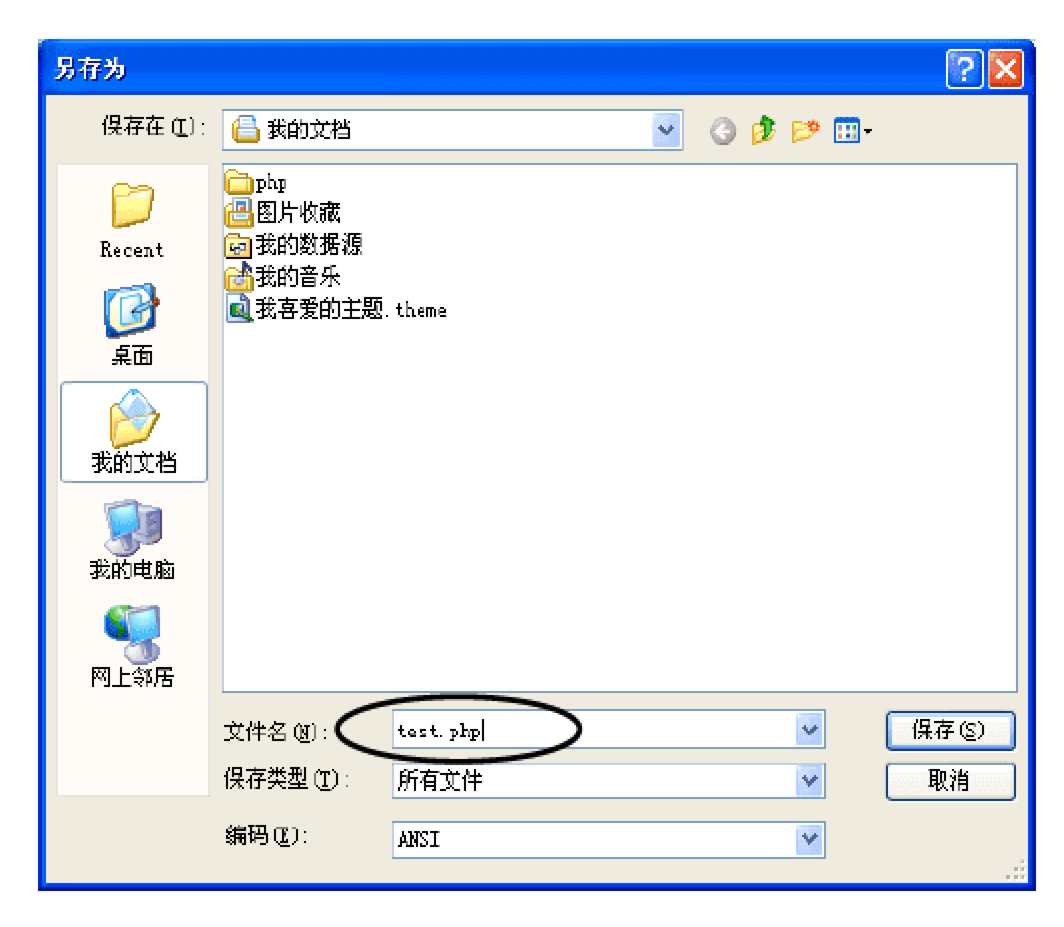


图 1-5 用记事本编写 PHP 代码并存储

当然，能开发 PHP 的工具绝非只有前面提到的几种，我们只是从每种类别的工具中选出具有代表性的来展示。这么多的工具，肯定有一款比较适合自己，就笔者而言，可视化开发工具推荐 FrontPage 或者 Dreamweaver，专业 PHP 开发工具推荐使用 Dev-PHP 与 PHPEdit，这些工具功能丰富，非常好用。当然，如果你舍得投入，也可以使用 Zend Studio 这款综合实力超群的工具。

1.5 案例 1：登录 Internet 了解 PHP 的应用情况

1.5.1 案例分析

本案例要求读者使用本地计算机到 Internet 上去搜索，浏览有关 PHP 的应用情况。读者只要使用任一台能连到 Internet 上的电脑就可以进行手动搜索了。

1.5.2 实施步骤

1. 打开 IE 或其他浏览器。
2. 登录到 http://www.baidu.com 或 http://www.google.com 搜索引擎。
3. 输入“PHP 应用”或其他读者想搜索的内容进行查找、浏览。

1.6　本 章 小 结

本章着重介绍了 PHP 嵌入式脚本的概念、来龙去脉、特点、在网络中的应用和开发工具。PHP 是免费的、嵌入式的、运行在服务器端的，发展到现在已经到了第 5 个版本 PHP 5，它功能强大、语法简洁、易于掌握，它在 Web 开发中占有重要位置，可以通过多种工具来进行 PHP 开发等等。读者通过本章的学习，应能对 PHP 脚本有一种初步的认识，对各类开发工具有大体的了解，并在以后的学习过程中熟练使用它们中的一种或几种。

1.7　练 习 题

1. 试描述 PHP 的概念。
2. 描述 PHP 的特点及与其他脚本的区别。
3. 描述目前常用的 PHP 开发工具名称及各自特点。

CHAPTER 2

预备知识：HTML 与 JavaScript



本章重点

- 常用 HTML 标记
- JavaScript 的使用

2.1　什么是 HTML

我们之所以要在学习 PHP 之前先学习一点 HTML 知识，是因为 PHP 本身就是一种嵌入式语言，它需要嵌入到 HTML 代码中来执行。因此在学习 PHP 之前首先掌握 HTML 是十分必要的。由于本书的重点是 PHP 开发，HTML 并不是我们要介绍的重点，因此本章中只是引导读者对 HTML 有一个基本认识。要深入学习 HTML，请参考其他专门介绍 HTML 的书籍。而对于那些已经很熟悉 HTML 的读者，则可以直接跳过这一章的学习。

2.1.1　HTML 的概念

HTML（HyperText Markup Language）即超文本标记语言，是 WWW 的描述语言。HTML 文档是由 HTML 命令组成的描述性文档，HTML 命令可以说明文字、图形、动画、声音、表格、链接等。HTML 的结构包括头部（Head）、主体（Body）两大部分，其中头部描述浏览器所需的信息，而主体则包含所要说明的具体内容。

HTML 是一个纯文本文件。创建一个 HTML 文档，只需要两个工具，一个是 HTML 编辑器，一个是 Web 浏览器。HTML 编辑器是用于生成和保存 HTML 文档的应用程序。Web 浏览器是用来打开 Web 网页文件，提供给我们查看 Web 资源的客户端程序，如 Internet Explorer、Netscape 等。

2.1.2　HTML 文档的结构

上面我们了解了 HTML 的概念，下边我们就以文档 2-1.html 为例子，讲述一下 HTML 文档的基本结构，使读者有一个宏观的认识。

程序清单　文档 2-1.html

```
<html>
<head>
    <title> HTML 文档范例</title>
</head>
<body>
 <center>
 <font size="6">这是一个 HTML 范例</font>
 </center>
</body>
</html>
```

打开 Windows 的记事本，把上面代码输入到记事本里面去（行号标记不要输入，后面例子与此例相同），然后保存文件，保存文件的界面如图 2-1 所示。

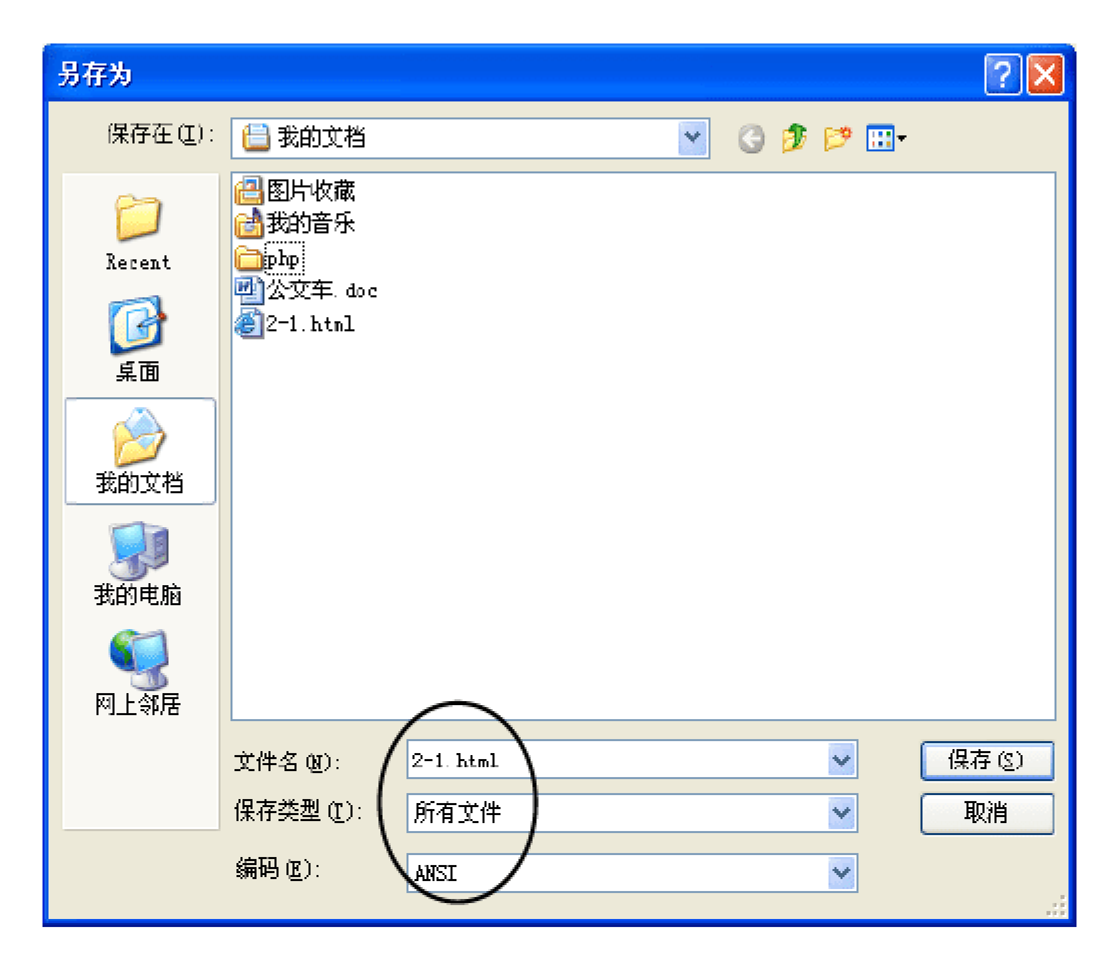


图 2-1　HTML 文档保存方式



输入代码完毕保存文件的时候，应当注意如下几点：

（1）文件名一栏，记得要把后缀名.html 或者.htm 加上；

（2）保存类型，一定选择所有文件，否则会默认保存为文本文档（后缀.txt）。

保存完毕之后，打开“我的文档”，就会看到目录下多了这样一个文件，这就说明文件正确保存了，双击用默认的 Internet Explorer 打开，网页效果如图 2-2 所示。

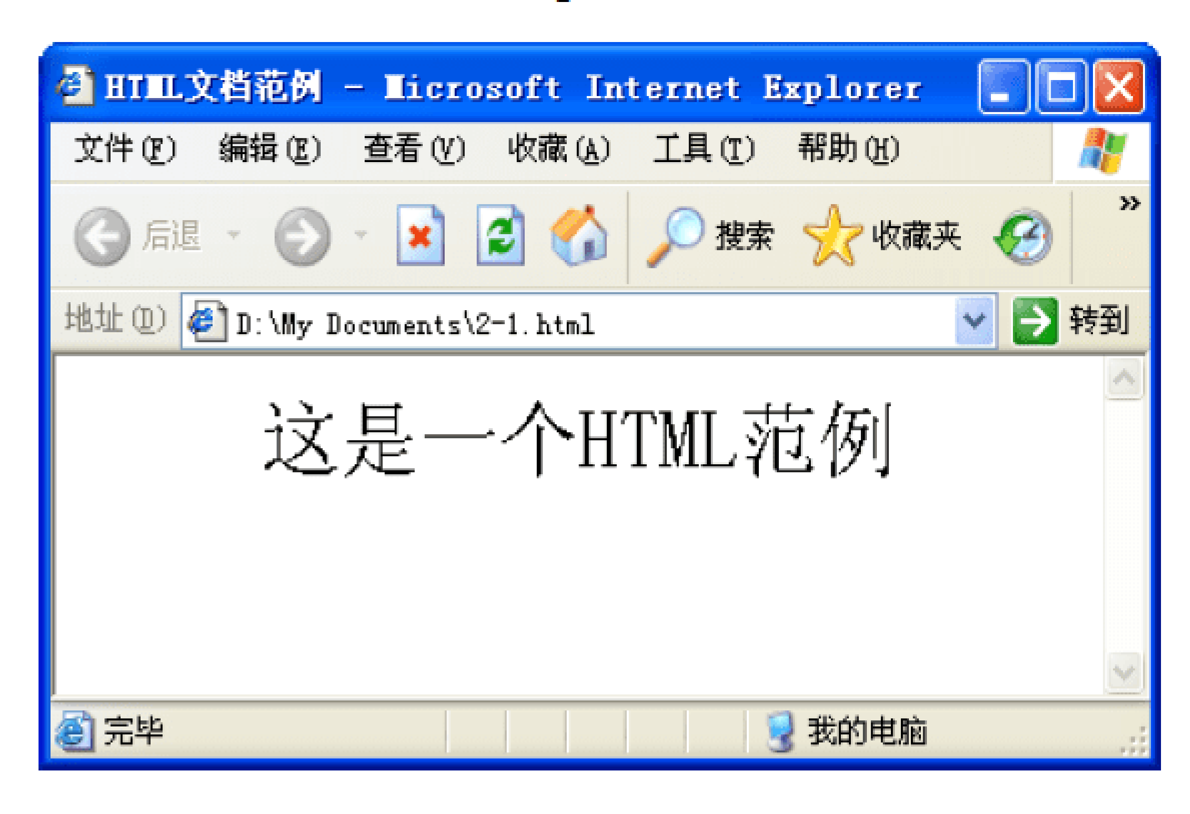


图 2-2　HTML 范例的执行结果

HTML 文档代码以标记<html>开始，以标记</html>结束，分文档头和文档体两部分。在文档头里，对这个文档进行了一些必要的定义，如网页标题，作者等。文档体中是要显示的各种文档信息。

<html></html>这一对标记在 HTML 文档的最外层，文档中的所有文本和 html 标记都包含在其中，表示该文档是以超文本标记语言（HTML）编写的。事实上，现在常用的 Web 浏览器都可以自动识别 HTML 文档，并不要求有<html>标记，也不对该标记进行任何操作，但是为了使 HTML 文档能够适应不断变化的 Web 浏览器，还是应该养成不省略这对标记的

良好习惯。<html>标记用于 HTML 文档的最前边，用来标识 HTML 文档的开始。而</html>标记恰恰相反，它放在 HTML 文档的最后边，用来标识 HTML 文档的结束，两个标记必须一块使用。

<head>和</head>构成 HTML 文档的开头部分，在此标记对之间可以使用<title>…</title>、<script>…</script>等标记，这些标记对都是描述 HTML 文档相关信息的，用以说明文档的标题和整个文档的一些公共属性。在浏览器窗口中，头部信息是不被显示在正文中的 ,若不需头部信息则可省略此标记，良好的习惯还是不省略。

<title>和</title>是嵌套在<head>头部标记中的，标记之间的文本是文档标题，它被显示在浏览器窗口的标题栏，如本例中标题栏显示为“HTML 文档范例”，见图 2-2 标题栏左上角。

<body>…</body>是 HTML 文档的主体部分，在此标记对之间可包含<p>…</p>、<h1>…</h1>、
、<hr>等等众多的标记，它们所定义的文本、图像等将会在浏览器窗口显示出来，<body>和</body>两个标记必须一块使用。

上面的这几对标记在文档中都是惟一的，<head>标记和<body>标记是嵌套在 HTML 标记中的。

2.1.3　HTML 的标记及属性

一个 HTML 文档是由一系列的元素和标记组成。元素名不区分大小写，HTML 用标记来规定元素的属性和它在文件中的位置。

刚刚接触 HTML 的读者，可能遇到的很大的障碍就是一些用“<”和“>”括起来的句子，我们称它为标记，是用来分割和标记文本元素的，以形成文本的布局、文字的格式及五彩缤纷的画面。标记通过指定某块信息为段落或标题等来标识文档的某个部件，属性是标记里的参数的选项。

HTML 的标记分单标记和成对标记两种。成对标记是由首标记<标记名> 和尾标记</标记名>组成的，成对标记的作用域只作用于这对标记中的文档。单独标记的格式<标记>，单独标记在相应的位置插入元素就可以了，大多数标记都有自己的一些属性，属性要写在首标记内，属性用于进一步改变显示的效果，各属性之间无先后次序，属性是可选的，属性也可以省略而采用默认值，其格式如下：

```
<标记名字 属性 1=值　属性 2=值　属性 3…>内容</标记名字>
```

例如，范例中的这是一个 HTML 范例，就是将文本“这是一个 HTML 范例”设置为六号字，size 即字号大小。属性可不使用引号引起来，但推荐使用，以养成良好的习惯。

提 示

> 输入首标记时，一定不要在“<”与标记名之间输入多余的空格，也不能在中文输入法状态下输入这些标记及属性，否则浏览器将不能正确的识别括号中的标记命令，从而无法正确的显示信息。

2.2 HTML 的文本标记及属性

在前面的一节中我们了解了 HTML 的概念、文档结构、标记及属性，对 HTML 有了初步认识，并成功编写了第一个 HTML 文档。在这一节里，开始学习 HTML 里的文本标记及项目符号。

HTML 的文本标记

下面将重点讲述 HTML 在文字版面设置方面常用到的一些标记，如换行、分段、文字缩排、特殊字符、注释以及文字的字体、大小、颜色等。这些标记都要写到文档主体<body>与</body>之间。首先看一个例子，HTML 文档 2-2.html，代码如下：

程序清单 文档 2-2.html

```
<html>
<head>
  <title>文本编辑</title>
</head>
<body>
这个不换行
这个不换行<br>
这是一个换行
<p align="center">
  <font size="5" face="隶书">这是个段落</font>
</p>
<blockquote>
  这里是缩排,<blockquote>再次缩排</blockquote>
</blockquote>
</body>
</html>
```

用记事本将上面代码保存为文档 2-2.html，再用 Internet Explorer 打开，效果如图 2-3 所示。

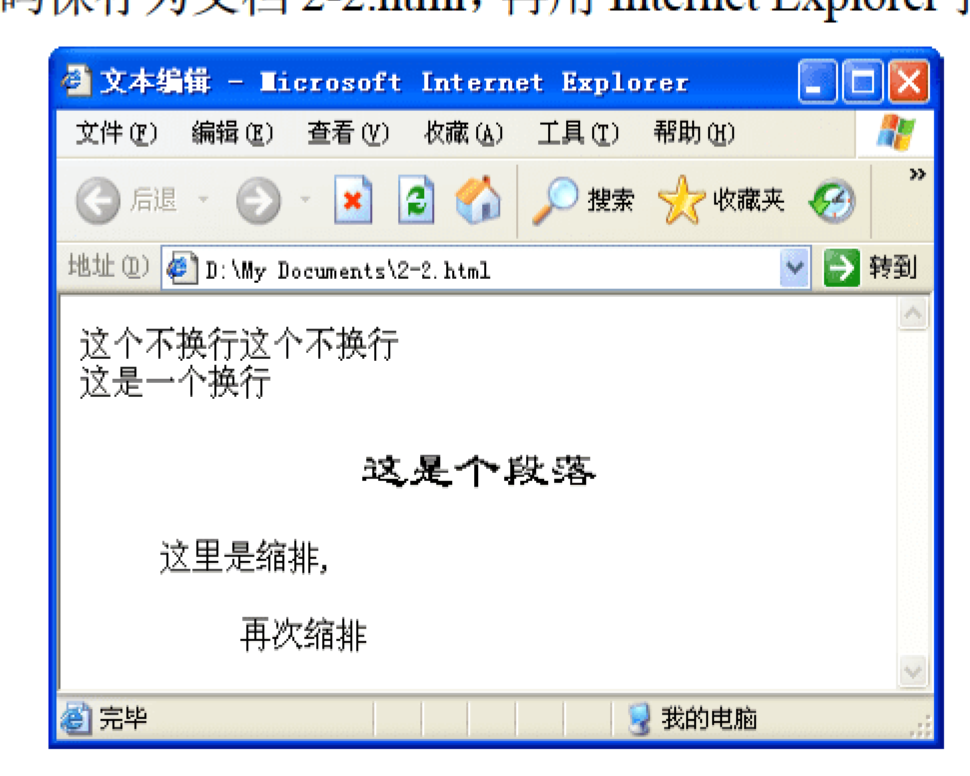


图 2-3 文字编辑界面

下面详细介绍一些常用标记及其属性。

1. 换行符

在 HTML 文本中的换行需要用本例中第 7 行的
表示，它是个单标记，没有属性，虽然在 HTML 3.0 之后可以写作
</br>，但没有实际意义。在 html 文档中的任何位置只要使用了
标记，当文档显示在浏览器中时，该标记之后的内容将显示在下一行。在 HTML 源代码里，对回车符是忽略的，所以要想换行就用
标记。

2. 分段符<p>

由本例中第 9 行的<p>标记所标识的文字，代表同一个段落的文字。不同段落间的间距等于连续加了两个换行符，用以区别文字的不同段落。它可以单独使用，也可以成对使用。单独使用时，下一个<p>的开始就意味着上一个<p>的结束。良好的习惯是成对使用。

分段符的写法是<p>，它有一个属性就是 align，有三种取值：left、center 和 right，分别代表居左、居中和居右，默认是居左。此标记写法为<p align= “值” >。具体用法见文档 2-2.html 与图 2-3。

3. 文字缩排<blockquote>

本例中第 12 行的<blockquote>标记可以用来建立一个引文，它特别适合较长文本的引用，引文显示时将会自动右移，左边有 2 个汉字的空白，加以区别。如图 2-3 所示，文本“这里是缩排”自动向右移动两个汉字的空白，而且缩排可以嵌套，文本“再次缩排”就是嵌套缩排的效果。

4. 预排版文字标记<pre>

要保留预排版文字的格式,就可以通过 2-3.html 源代码中第 2 行的<pre>标记来实现。方法是把制作好的文字排版内容前后分别加上首标记<pre>和尾标记</pre>。

HTML 文档 2-3.html（首、尾标记已省去，只留主体标记部分）的主体代码如下：

程序清单　文档 2-3.html

```
1:<body>
2:<pre>
3:海纳百川
4: 有容乃大
5:HTML 预排版文字格式
6:</pre>
7:<!--这是保持预排版的标记-->
8:</body>
```

运行后效果如图 2-4 所示。

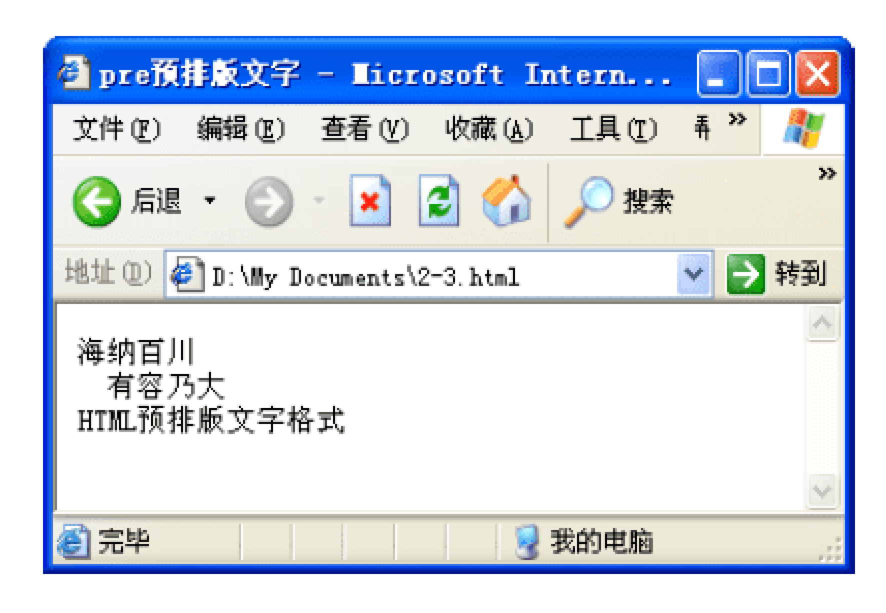


图 2-4　pre 预排版文字

5. 居中对齐标记<center>

文本使用<center>标记标注后可在页面中居中显示，<center>是成对标记，在需要居中的内容部分开头处加<center>，结尾处加</center>。

具体用法见程序清单文档 2-4.html 的第 6 行和第 11 行，效果图如 2-5 所示。

6. 注释标记<!-->

在 HTML 文档中可以加入相关的注释标记，便于查找和记忆有关的文档内容和标识，这些注释内容并不会在浏览器中显示出来。

注释标记的格式是：

```
<!--这里是注释的内容-->
```

例子请见 2-3.html 文档中的第 7 行及图 2-4。

7. 字体的设置…

上面几项学习了排版中用到的一些标记，接下来学习字体标记的属性设置，这是文字编辑的重点，也是在网页中应用最普遍的功能，应该重点掌握。首先看一下下面的图 2-5。

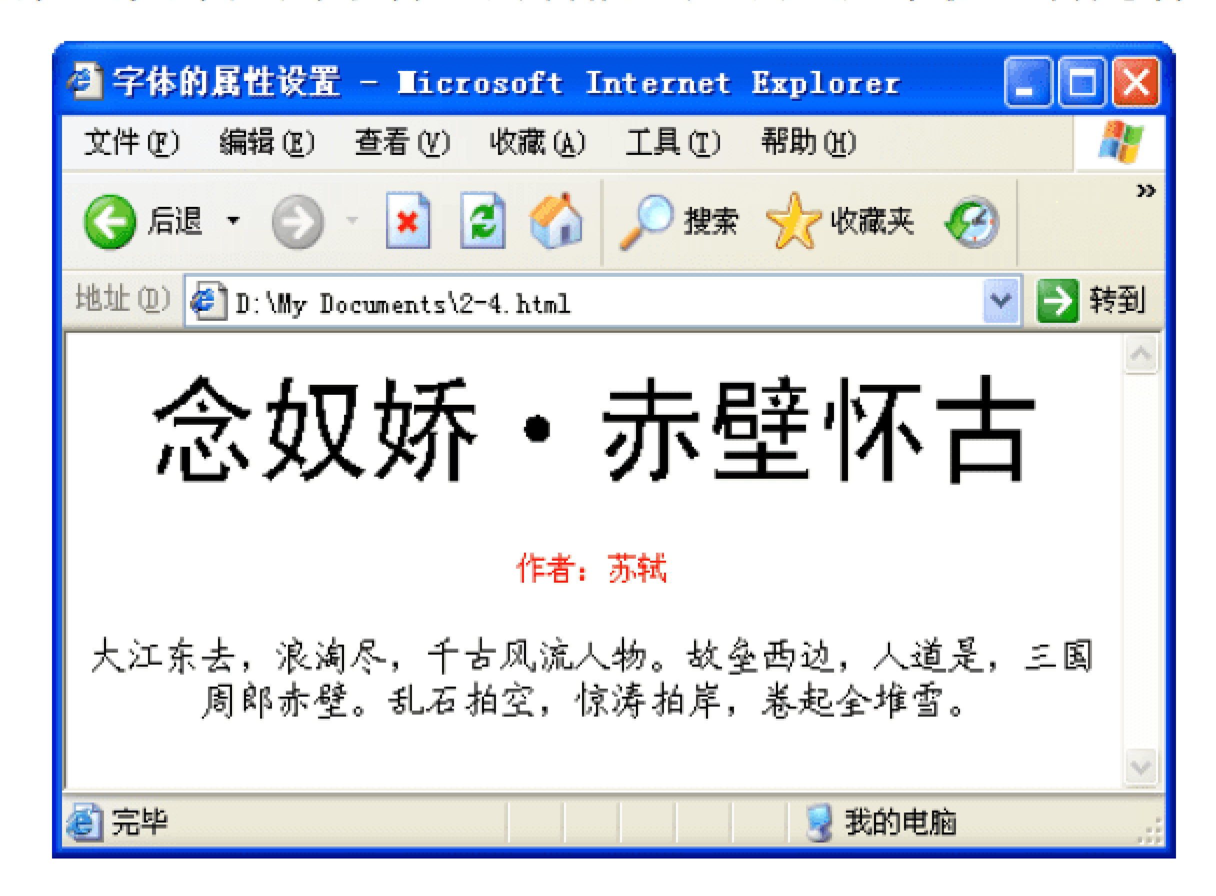


图 2-5　文字居中及字体标记

通过这个图就能看出来，这里面字体大小、字体是不一样的，这首词的词牌字体最大、黑体，作者字体最小、宋体，词内容较小、楷体。现在我们就看一下源文件，代码如下：

程序清单　文档 2-4.html

```
1:<html>
2:<head>
3:   <title>字体的属性设置</title>
4:</head>
5:<body>
6:<center>
7:   <font size="7" face="黑体">念奴娇·赤壁怀古</font><p>
8:   <font size="2" face="宋体" color="red">作者：苏轼</font><p>
9:   <font size="3" face="楷体_GB2312">大江东去，浪淘尽，千古风流人物。
10:故垒西边，人道是，三国周郎赤壁。乱石拍空，惊涛拍岸，卷起千堆雪。</font>
11:</center>
12:</body>
13:</html>
```

字体的设置，我们是通过本例中第 7~10 行的这对标记对来实现的。font 标记有三个属性：face、size 和 color，分别代表着字体、字号、颜色。格式为：

```
<font face="值 1" size="值 2" color="值 3">文字</font>
```

下面用表 2-1 说明文字的属性及默认值。

表 2-1　<font>标记属性功能及取值

属性	使用功能	默认值
face	设置文字使用的字体	宋体
size	设置文字的大小	3
color	设置文字的颜色	黑色

face 属性的取值有宋体、楷体_GB2312、隶书、黑体等等，视操作系统支持的字体库而定，如果采用的字体在操作系统字体库里不存在，则以默认的宋体来取值。

size 为字号属性，取值为从 1 到 7，数值越大，字体越大。

color 为颜色属性，取值有两种。一种是颜色的英文名称，如 red、black、green、blue、pink、yellow，分别代表红色、黑色、绿色、蓝色、粉红、黄色。第二种是 RGB 颜色，格式为“#nnnnn”，n 为十六进制数，取值从 0 到 F。如果读者对 RGB 颜色代码感兴趣，可查阅相关资料，这里不再详述。

8. 标题文字标记<hn>

<hn>标记用于设置网页中的标题文字，被设置的文字将以黑体或粗体的方式显示在网页中。

标题标记的格式：

```
<hn align=参数>标题内容</hn>
```

文档 2-5.html 是标题文字标记应用实例，效果如图 2-6 所示。

程序清单　文档 2-5.html

```
<body>
<h1 align="center">这是 h1 标题字</h1><p>
<h2 align="right">这是 h2 标题字</h2><p>
<h3 align="left">这是 h3 标题字</h3><p>
</body>
```

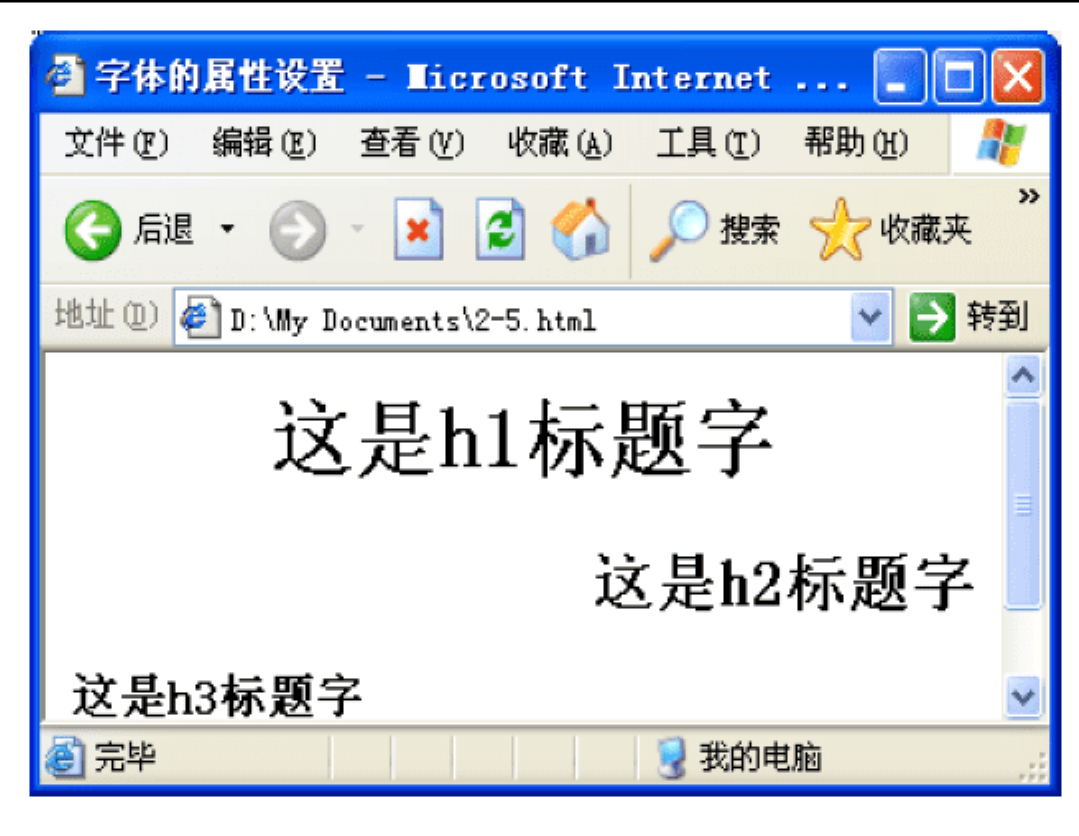


图 2-6　标题字的设置

说 明

<hn>标记是成对出现的，共分为六级，在<h1>…</h1>之间的文字就是第一级标题，是最大最粗的标题，<h6>…</h6>之间的文字是最后一级，是最小最细的标题文字。align 属性用于设置标题的对齐方式，其取值为 left（左），enter（中），right（右）。<hn>标记本身具有换行的作用，标题总是从新的一行开始。如图 2-6 所示。

9. 特定文字样式标记

在有关文字的显示中，常常会使用一些特殊的字形或字体来强调、突出、区别，以达到提示的效果。如粗体标记…、斜体标记<i>…</i>、下划线标记<u>…</u>。当然还有一些其他的样式标记，这里不做详细介绍。

2.3　插入图片和超链接

2.3.1　HTML 中图片的插入

图像可以使 html 页面美观生动且富有生机，再制作精美的网页如果只有文字而没有图像的话将失去许多活力。这一节就来学习 HTML 中图像的插入，浏览器可以显示的图像格式有 jpg、bmp、gif 等。

首先，我们看一个插入图片的网页，如图 2-7 所示。



图 2-7　图片与文字的混排

图 2-7 页面的文档 2-8.html 的主体源代码如下：

程序清单　文档 2-8.html

```
1:<body>
2:<img src="xiangrikui.jpg" width="150" height="120" border="0"
3:align="left" hspace="20" vspace="2">
4:    这是一片向日葵的世界，在这里，你将享受到一份难得的宁静，
  你甚至能够听到向日葵生长的声音和它们向着太阳转头的快乐。
5:</body>
```

1. 插入图片代码分析

让我们分析一下这段主体代码，见本例中第 2 行，<img src="xiangrikui.jpg" width="150" height="120" border="0" align="left" hspace="20" vspace="2">就是插入图片的命令以及属性的设置，后边的四个" "为四个英文空格，正好是两个汉字的空白，在后面就是文字了，我们采用的是默认的设置。

网页中插入图片用单标记<img>，当浏览器读取到<img>标记时，就会显示此标记所设定的图像。如果要对插入的图片进行修饰时，仅仅用这一个属性是不够的，还要配合其他属性来完成。标记<ing>的属性如表 2-2 所示。

表 2-2　标记<img>的属性

属性	描述
src	图像的 url 路径
alt	提示文字，当浏览器无法显示图片时，将显示此提示文字
width	宽度，通常只设为图片的真实大小以免失真
height	高度，通常只设为图片的真实大小以免失真
align	图像和文字之间的排列属性，这里设的是左对齐方式
border	边框，以像素为单位
hspace	水平间距
Vspace	垂直间距

2. img 标记的属性

表 2-2 列举出了<img>里面所含有的属性，当然，src 属性是必须含有的，它的取值是图像的路径和图像名字，其他的都是可选的。下面分析这几个属性的用法。

（1）src 属性：这个属性的取值是图像文件的文件名，包括路径，这个路径可以是相对路径，也可以是绝对路径，实际上就是通过路径将图形文件嵌入到文档中。要想对路径的知识有全面的了解，请先学习 2.3.3 节的“链接路径”部分。此例中，src 的取值为“xiangrikui.jpg”，这是图像文件的名字，它和文档 2-8.html 在同一目录下。

（2）alt 属性：这是图片的提示文字，当鼠标放到图片上静止 1 秒钟时，提示文字就会出现在鼠标指针下方。当图片由于各种原因无法正常显示时，提示文字就会自动出现在图片的位置。读者可以自己试验一下 alt 的用法，alt 属性在图片中应用很广泛。

（3）width 和 height 属性：分别代表图像文件的宽度和高度，若只设置了其中一个值，则图片按长宽比自动调整另外一个值，若两属性都被赋值且不符合长宽比，则会出现图片比例失真的情况，当设置值比图像原始尺寸大的时候，会出现像素失真。建议大家使用此属性的时候使用默认值或只设置一个属性。

（4）border 属性：为边框的像素值，本例中 border 设置为 1，代表边框为 1 个像素线宽，最小值可设为 0。

（5）align 属性：图像和文字的之间的对齐是通过 align 属性来设定的，align 的对齐方式有两种：即绝对对齐和相对对齐。绝对对齐方式的效果和文字一样，只有三种：居中（middle）、居左（left）、居右（right）。相对文字对齐方式是指图像与一行文字的相对位置。此例里面的 align=left，即图像左对齐，下面的文章就在图片的右边显示，完成排版。Align 属性的设定值如表 2-3 所示。

表 2-3 对齐属性 align 的设定

属性值	对齐方式
top	上对齐
middle	居中对齐（垂直方向）
center	居中对齐（水平方向）
bottom	下对齐
left	左对齐
right	右对齐

（6）hspace 和 vspace 属性：hspace 是水平间距，即在图像的左侧和右侧留有给定像素值的空白，如此例里的 hspace=20，代表图像左、右间距都是 20 像素，vspace 是垂直间距，即图片的上方和下方留有给定像素值的空白，本例中 vspace 设置为 2 像素。

2.3.2 HTML 中背景图片的使用

网站上经常有一些背景图片，背景图片用的好，能使网页更加漂亮、更能突出主题。

设置背景图片的方法是，在<body>标记里加入属性 background= “图片地址”。打开 HTML 编辑工具，输入以下主体代码：

程序清单 文档 2-9.html

```
<body background="bg.jpg">
  <p>
  <center>
      <font size=6 face="黑体">背景图片的插入方法</font>
  </center>
</body>
```

执行效果如图 2-8 所示。

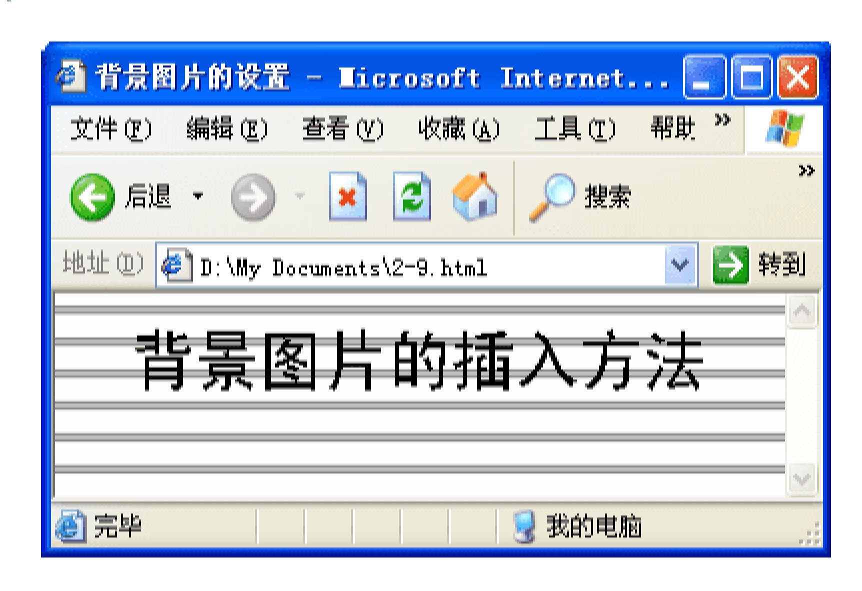


图 2-8 背景图片的显示效果

在图 2-8 所示网页中，有很多的灰条，但其实图片文件 bg.jpg 只是一个灰度条，可见，当设置为背景图片后，网页窗口部分就会把图片平铺到整个的页面上，所有其他的内容都会在背景之上显示。

2.3.3 HTML 中的超链接

HTML 文档最重要的应用之一就是超链接，超链接是一个网站的灵魂，Web 上的网页是互相链接的，单击超链接的文本或图形就可以链接到其他页面。超链接具有的链接能力，可层层链接相关文件，这种具有超级链接能力的操作，即称为超级链接。超级链接除了可链接文本外，也可链接各种媒体，如声音、图像、动画，通过超级链接我们可享受丰富多彩的多媒体世界。

首先来看一个含有超级链接的网页，如图 2-9 所示。

在这个网页中，有些文本是蓝色、带有下划线的，这就是超级链接的标记，这个例子的主体代码如下：

程序清单 文档 2-10.html

```
<body>
常用站点：<br>
<a href="http://www.sina.com.cn" title="新浪" target="_blank">新浪
</a>  
```

```
5:<a href="http://www.sohu.com" >搜狐</a>  
6:<a href="http://www.163.com">网易</a>
7:<p>
8:本书中的几个例子：<br>
9:<a href="2-1.html">2-1.html</a>  
10:<a href="2-2.html">2-2.html</a>  
11:<a href="2-3.html">2-3.html</a>
12:</body>
```

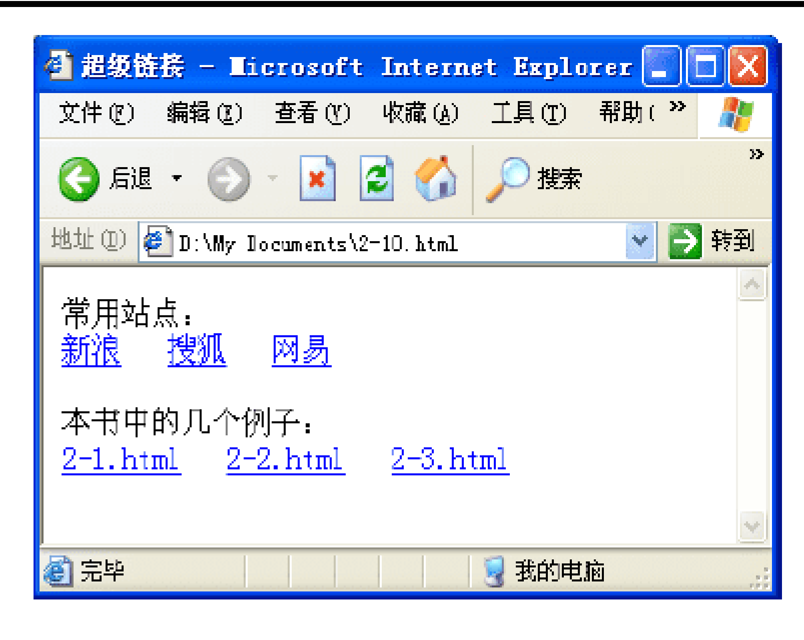


图 2-9 文本超级链接

下面对这段代码进行分析。

1. 超链接标记<a>/

建立超链接的标记对为本例中第 3 行的<a>和，格式为：

```
<a href="资源地址" target="窗口名称" title="提示文字">超链接名称</a>
```

href 属性定义了这个链接所指的目标地址。目标地址是最重要的，一旦路径上出现差错，该资源就无法被访问。

target 属性用于指定打开链接的目标窗口，共有 4 种打开方式（见表 2-4），其默认方式是原窗口。

表 2-4 目标窗口的属性

属性值	描述
_parent	在上一级窗口中打开，一般使用分帧的框架页会经常使用
_blank	在新窗口打开
_self	在同一个帧或窗口中打开，这项一般不用设置
_top	在浏览器的整个窗口中打开，忽略任何框架

title 属性用于指定指向链接时所显示的标题文字。

2. 链接地址

每一个文件都有自己的存放位置和路径，理解一个文件到要链接的那个文件之间的路径关系是创建链接的根本。链接可以分以下两种：

（1）绝对路径

绝对路径：包含了标识 Internet 上的文件所需要的所有信息。文件的链接是相对原文档而定的。包括完整的协议名称，主机名称，文件夹名称和文件名称。

其格式为：

通讯协议://服务器地址:通讯端口/文件位置……/文件名

比如上例中第 2 行到第 5 行里面的“新浪”就链接到 http://www.sina.com.cn，这是一个绝对路径。

（2）相对路径

相对路经是以当前文件所在路径为起点，进行相对文件的查找。此时，浏览器链接的目标文档处在同一服务器或同一文件夹下。如本例中第 9 行到第 11 行。

如果链接到同一目录下，则只需输入要链接文件的名称。

要链接到下级目录中的文件。只需先输入目录名，然后加“/”，再输入文件名。

要链接到上一级目录中文件，则先输入“../”，再输入文件名。

2.4　表格的使用

表格标记对于制作网页是很重要的，可以方便灵活的排版，很多动态大型网站都是借助表格排版，表格可以把相互关联的信息元素集中定位，使浏览页面的人一目了然。所以说要制作好网页，就要学好表格。

读者可能都用过 Word 字处理软件和 Excel 电子表格，这两款软件都有很强大的表格功能，今天来学习 HTML 里生成表格的标记，首先看一个含有表格的网页，如图 2-10 所示。

本网页的主体代码为：

程序清单　文档 2-11.html

```
<body>
<table width="300" height="100" align="center" bgcolor="pink" border="1"
  bordercolor="black">
 <tr> <td colspan="3" align=center>登鹳雀楼</td>  </tr>
 <tr align="center">
  <td bgcolor="yellow">白<p>日<p>依<p>山<p>尽</td>
 <td width="210">
    <img src="bairiyishanjin.jpg" width=200>
  </td>
  <td bgcolor="yellow">黄<p>河<p>入<p>海<p>流</td>
</tr>
</table>
</body>
```

可以看到，这个网页中的排版，是原来所学的文字排版所不能实现的，这正是表格在

排版中体现的作用。

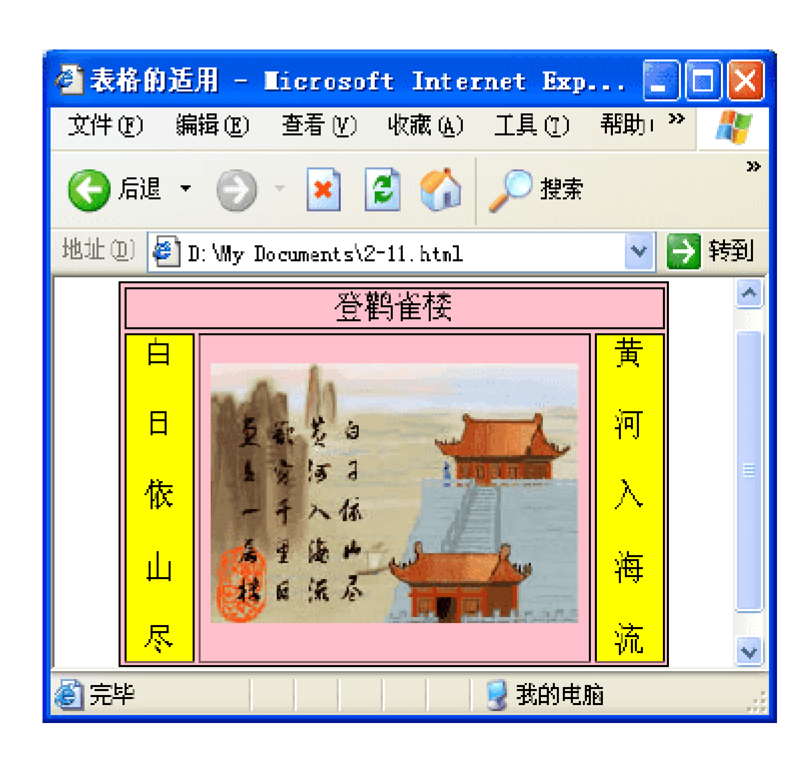


图 2-10　HTML 中的表格

2.4.1　定义表格的基本标记

在 HTML 文档中，表格是通过<table>、<tr>、<td>标记来完成的，如表 2-5 所示。

表 2-5　表格标记

标记	描述
<table>…</table>	定义一个表格开始和结束
<tr>…</tr>	定义一行，一组行标记内可以建立多组由<td>标记所定义的单元格
<td>…</td>	定义单元格，一组<td>标记建立一个单元格，<td>标记必须放在<tr>标记内

2.4.2　表格标记及其属性

1. 表格标记<table>属性

表格标致记<table>最常用的属性如表 2-6 所示。

表 2-6　<table>标记属性

属性	描述
width	表格的宽度，可用像素或比分比表示
height	表格的高度，可用像素或百分比表示
align	表格在页面的水平摆放位置
background	表格的背景图片
bgcolor	表格的背景颜色
border	表格边框的宽度（像素为单位），可设置为 0
bordercolor	表格边框颜色
bordercolorlight	表格边框明亮部分的颜色

（续表）

属性	描述
bordercolordark	表格边框昏暗部分的颜色
cellspacing	单元格之间的间距
cellpadding	单元格内容与单元格边界之间的空白距离的大小

如在文档 2-11.html 的第 2 行，设置的 table 标记为：

```
<table width="300" height="100" align="center" bgcolor="pink" border="1" bordercolor="black">
```

即代表表格宽度是 300 像素，高度是 100 像素，表格在水平方向的位置是居中，背景颜色是粉红色，表格边框是 1 像素，边框颜色是黑色。这个表格显示的是双线边框，这是因为默认的 cellspacing 属性（单元格间距）为 1 像素。其他的属性，各位读者可以自己试验一下，多感受这些属性给表格带来的变化。

2. 行标记<tr>…</tr>

表格是由行和列（单元格）组成的，一个表格由几行组成就要有几个行标记<tr>，如文档 2-11.html 的第 3 行，行标记用它的属性值来修饰，属性都是可选的。详细属性如表 2-7 所示。

表 2-7　<tr>标记的属性

属性	描述
align	行内容的水平对齐
valign	行内容的垂直对齐
bgcolor	行的背景颜色
bordercolor	行的边框颜色
bordercolorlight	行的亮边框颜色
bordercolordark	行的暗边框颜色

文档 2-11.html 中用到了<tr align=center>，这说明将这一行的所有单元格的水平对齐方式设置为居中。其他的属性在表 2-7 中已经给出，读者可以自己练习一下。

3. 单元格标记<td>…</td>

<td>是插入单元格的标记，这个标记必须嵌套在<tr>标记内，是成对出现的。在<td>…</td>之间的文字是该单元格中的具体数据内容，<td>属性设定见表 2-8。

表 2-8 <td>的属性

属性	描述
width/height	单元格的宽和高，接受绝对值（如 80）及相对值（如 80%）。
colspan	单元格向右打通的栏数
rowspan	单元格向下打通的列数
align	单元格内字画等的摆放位置（水平），可选值为：left,center,right
valign	单元格内字画等的摆放位置（垂直），可选值为：top,middle,bottom
bgcolor	单元格的底色
bordercolor	单元格边框颜色
bordercolorlight	单元格边框向光部分的颜色
bordercolordark	单元格边框背光部分的颜色
background	单元格的背景图片

一些名称相同的属性在前面已经介绍过了，这里不再赘述，主要介绍一下 colspan 和 rowspan 属性，这两个属性的用处是合并单元格。表格的第一行，本来是三个单元格的位置，在文档 2-11.html 的代码里设置<td>标记的 colspan=3 后，就相当于把此单元格跨越了 3 个列的位置，把后面两个单元格“吞并”了。rowspan 的用法和 colspan 一样，不同的是向下合并单元格，读者可以动手试一下。

2.4.3 表格的嵌套

在 HTML 页面中，使用表格排版是通过嵌套来完成的，即一个表格内部可以嵌套另一个表格，用表格来排版页面的思路是：由总表格规划整体的结构，由嵌套的表格负责各个子栏目的排版，并插入到表格的相应位置，这样就可以使页面的各个部分互不冲突，看上去清晰整洁。

2.5 创建表单

表单是 HTML 中很重要的一部分知识，它在 Web 网页中用来给访问者填写信息，从而能采集客户端信息，使网页具有交互的功能。一般是将表单设计在一个 HTML 文档中，当用户填写完信息后做提交（submit)操作，于是表单的内容就从客户端的浏览器传送到服务器上，经过 Web 服务器处理后，再将用户所需信息传送回客户端的浏览器上，这样网页就具有了交互性。这里只讲怎样使用 HTML 标记来设计表单，涉及到交互和收集变量的知识在后边学 PHP 的时候会重点讲述。

首先看一个有表单的网页，如图 2-11 所示。

图 2-11 对应的主体源代码为：

程序清单　文档 2-12.html

```
<body>
<form action="2-12.php" method="POST" name="myform">
名字：<input type=text name="name" size=15 maxlength=8><br><!--文本输入框-->
性别：男<input type="radio" name="sex" value="男" checked>  
女<input type="radio" name="sex" value="女"><br><!--单选按钮-->
你的爱好：看书<input type="checkbox" name="like" value="看书"> 
        足球<input type="checkbox" name="like" value="足球" checked> 
        音乐<input type="checkbox" name="like" value="音乐"> 
        爬山<input type="checkbox" name="like" value="爬山"><!--多选按钮-->
<br>
你最喜欢的颜色：
    <select name="color" size="1">
     <option name="red">红色
     <option name="green" selected>绿色
     <option name="blue">蓝色
   </select><!--下拉列表框-->
<br>
个人介绍：
<textarea name="jieshao" cols="30" rows="5"></textarea><!--多行文本框-->
<p>
<center>
<input type="submit" value="提交">  
<input type="reset">
</center>
</form>
</body>
```

程序运行后，界面如 2-11 所示。

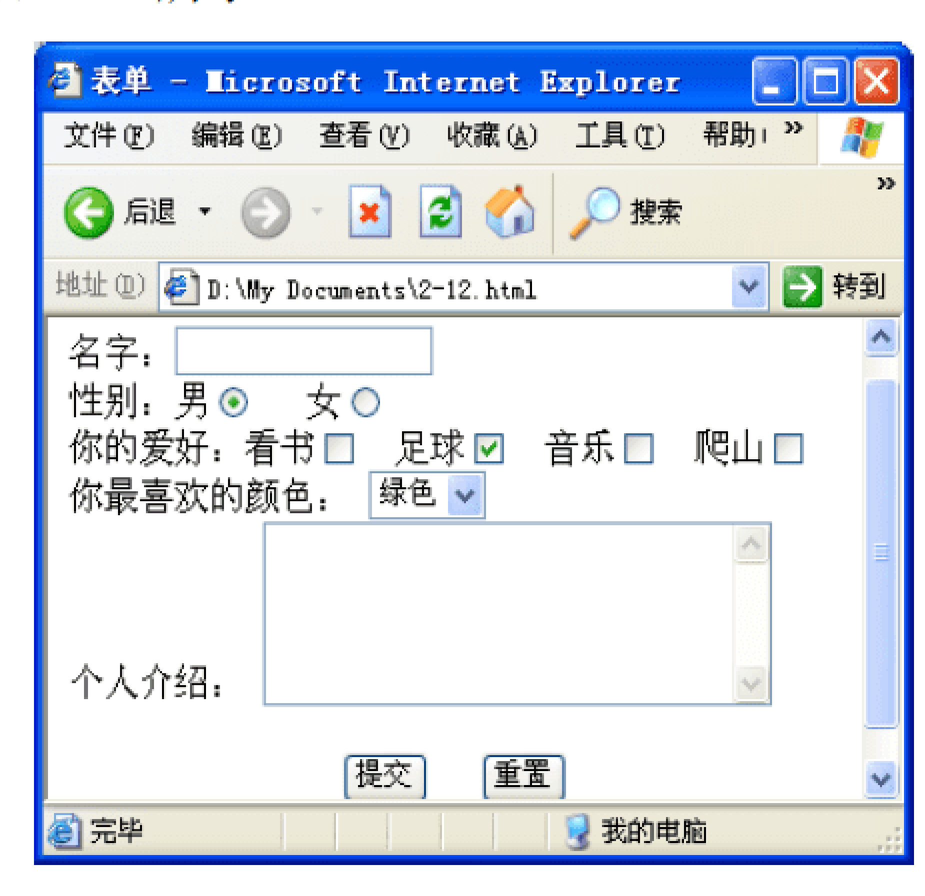


图 2-11　HTML 中的表单及控件

2.5.1 HTML 中的表单标记<from>…<form>

从面向对象的编程思想来看，表单是由窗体和控件组成的，一个表单一般应该包含用户填写信息的输入框和提交按钮等，这些输入框和按钮叫做控件，表单很像容器，它能够容纳各种各样的控件。

表单由<form>…</form>标记对来创建。<form>…</form>定义表单的开始和结束之间的一切定义都属于表单的内容。<form>标记具有 name、action、method 和 target 属性。表单标记的格式为：

```
<form action="url" method=get|post name="myform" target="_blank">…</form>
```

其中属性的作用如下：

（1）name 是表单的名称，在同一个页面中，应该有一个惟一的命名，这个在运用 JavaScript 处理表单的时候会用的到。

（2）action 的值是处理程序的程序名（包括网络路径：网址或相对路径），如：<form action="用来接收表单信息的 url">，如果这个属性是空值（""），则当前文档的 url 将被使用。当用户提交表单时，服务器将执行 action 值对应的程序（如 PHP 程序）。

（3）method 属性用来定义处理程序从表单中获得信息的方式，可取值为 get 或 post 的其中一个。get 方式是处理程序从当前 HTML 文档中获取数据，这种方式传送的数据量是有所限制的，一般限制在 1KB（255 个字节）以下。post 方式传送的数据量比较大，它是把数据人当前的 HTML 文档传送给处理程序，传送的数据量要比使用 get 方式时大的多。

（4）target 属性用来指定目标窗口或目标帧。可取值为当前窗口_self，父级窗口_parent，顶层窗口_top，空白窗口_blank。

2.5.2 表单中的输入标记<input>

在 HTML 语言中，标记<input>具有重要的地位，如文档 2-12.html 中第 3 行，将浏览器中的控件加载到 HTML 文档中。<input>是单个标记，没有结束标记。<input type="">标记定义了一个用户输入区，用户可在其中输入信息。此标记必须放在 <form></form>标记对之间。<input type="">标记共提供了 9 种类型的输入区域，具体是哪一种类型由 type 属性来决定，type 属性的取值及对应的实际含义如表 2-9 所示。

表 2-9　type 属性值定义

type 属性	输入区域类型及说明
<input type="text" size="" maxlength="">	单行的文本输入区域，size 与 maxlength 属性用来定义此种输入区域显示的尺寸大小与输入的最大字符数
<input type="submit">	提交到服务器的按钮，当这个按钮被单击时，就会链接到表单 form 属性 action 指定的 url 地址
<input type="reset">	重置按钮，单击该按钮可将表单内容全部清除，重新输入数据
<input type="button">	普通按钮，当这个按钮被单击时，就会调用 JavaScript 事件

（续表）

type 属性	输入区域类型及说明
<input type="checkbox" checked>	一个复选框，checked 属性用来设置该复选框默认状态时是否被选中
<input type="hidden">	隐藏区域，用户不能在其中输入，用来预设某些要传送的信息
<input type="image" src="url">	使用图像来代替 Submit 按钮，图像的源文件名由 src 属性指定，用户单击后，表单中的信息和单击位置的 X、Y 坐标一起传送给服务器
<input type="password">	输入密码的区域，当用户输入密码时，区域内将会显示“*”号
<input type="radio">	单选按钮类型，checked 属性用来设置该单选框默认状态时是否被选中

不同类型的输入区域有一个公共的属性 name，此属性指定给每一个输入区域一个名字。名字与输入区域是一一对应的，即一个输入区域对应一个名字，当 type=checkbox 时除外，此时多选按钮各个输入区域的 name 属性都设置为同一个。服务器就是通过调用某一输入区域的名字的 value 值来获得该区域的数据的。value 属性是另一个公共属性，它用来指定输入区域的默认值。如源代码中的文本框、单选按钮、多选按钮，大家可以动手实验一下。

2.5.3　表单中的下拉列表框<select>…</select>

文档 2-12.html 中第 9 行的<select>…</select>标记对创建了一个菜单下拉列表框。此标记对位于<form>…</form>标记对之间。<select>具有 multiple、name 和 size 属性。multiple 属性不用赋值，直接加入标记中即可使用，加入此属性后列表框就成可多选的了；name 是此列表框的名字，它与上边讲的 name 属性作用是一样的；size 属性用来设置列表的高度，默认状态下的值为 1，若没有设置 size 属性，显示的将是一个弹出式的列表框。

<option>标记用来指定列表框中的一个选项，它位于<select>…</select>标记对之间。此标记具有 selected 和 value 属性。selected 用来指定默认的选项；value 属性用来给<option>指定的那一个选项赋值，这个值是要传送到服务器上的，服务器通过调用<select>区域名字的 value 属性来获得该区域选中的数据项。

在图 2-11 中，我们有一个“你喜欢的颜色”的下拉单选列表框，大家可以动手实验。

2.5.4　表单中的多行文本框<textarea>…</textarea>

文档 2-12.html 中第 12 行的<textarea>…</textarea>用来创建一个可以输入多行文字的多行文本框，此标记对位于<form>…</form>标记对之间。<textarea>的几个常用属性如下：

（1）name：文本框的名称，传递文本内容所用。
（2）cols：文本框的宽度，赋值以字符为单位。
（3）rows：文本框的列数，即其高度，赋值以字符为单位。

图 2-11“个人介绍”一栏用的就是多行文本框。

至此，HTML 的常用的标记和用法已做了些讲述，限于篇幅，对 HTML 的讲述和举例不可能面面俱到。读者应该对 HTML 有了初步的认识，如果需要更加深入的了解，可以参

考其他的书籍。下面要讲的 JavaScript，是能更好的帮助我们进行 Web 开发的好工具。

2.6 JavaScript 简介

JavaScript 是一种轻型的、解释性的程序设计语言，具有面向对象的能力。该语言的通用核心已经嵌入了 Netscape、Internet Explorer 和其他的 Web 浏览器中，它使 Web 程序设计增色不少。JavaScript 的客户端版本把可执行的内容添加到了网页中，这样一来，网页就不是静态的 HTML 页面了，而是包含了与用户进行交互的程序、控制浏览器的程序以及动态创建 HTML 内容的程序。

通过 JavaScript 可以做到响应用户的需求事件（如表单的输入）。这样，当使用者输入一项信息时，它不需要通过网络传送到服务器端进行处理再传回来的过程，而可以直接在客户端进行事件的处理，也可以想像成有一个可执行程序在客户端上执行一样（但这种执行程序的功能是有限的，它对客户端电脑的控制力很差）。

2.7 JavaScript 语法

2.7.1 在什么地方插入 JavaScript

使用标记<script>…</script>，可以在 HTML 文档的任意地方插入 JavaScript，甚至在<HTML>文档之前插入也不成问题。

一般的做法是将其插入到<head>…</head>之间，这里也推荐这种做法。

插入 JavaScript 的基本格式：

```
<script language="JavaScript">
<!--
(JavaScript 代码)
//-->
</script>
```

“<!--”与“//-->”的作用，是让不支持<script>标记的浏览器忽略 JavaScript 代码，一般可以省略，因为主流的浏览器无一例外的都支持<script>。第 4 行前边的双反斜杠“//”是 JavaScript 里的注释标记。

另外一种插入 JavaScript 的方法，是把 JavaScript 代码写到另一个文件当中（此文件通常应该用“.js”作扩展名），然后把格式为“<script src="javascript.js"></script>”的标记嵌入到 HTML 文档中。注意，一定要用“</script>”标记。

说 明

相对于<script>标记，还有一个<server>标记。<server>标记所包含的，是服务器端的脚本。本书只讨论客户端 JavaScript，也就是用<script>标记包含的脚本。

2.7.2　JavaScript 中的变量

1. JavaScript 基本语法

每一句 JavaScript 都有类似于以下的格式：

<语句>;

其中，分号“;”是 JavaScript 语言作为一个语句结束的标识符。

语句块的概念：语句块是用大括号“{ }”括起来的一个或 n 个语句。在大括号里边是几个语句，但是在大括号外边，语句块是被当作一个语句的。语句块是可以嵌套的，也就是说，一个语句块里边可以再包含一个或多个语句块。

2. 变量

变量是用于存储某种/某些数值的存储器。所存储的值，可以是数字、字符等数据。

（1）变量的命名

在 JavaScript 中，变量名大小写所表示的意义是不同的，但有一些浏览器并不区分，不过对变量的命名应该养成一个好习惯，以便自己和他人识别，另外需注意，变量名必须以字母或下划线开始。

（2）变量的声明与类型

变量需要声明，没有声明的变量不能使用，否则会出错：“未定义”。变量声明的格式如下：

```
var 变量名 [= <值>];
```

变量可以用的数据类型有如下几种：

- **整型**：只能储存整数。
- **浮点型**：即“实型”，能储存小数。
- **字符串型**：是用引号“" "”、“' '”包起来的零个至多个字符。用单引号还是双引号没有区别。
- **布尔型**：常用于判断，只有两个值可选：true（表“真”）和 false（表“假”）。

由于 JavaScript 对数据类型的要求不严格，一般来说，声明变量的时候不需要声明类型。而且就算声明了类型，在过程中还可以给变量赋予其他类型的值。可以通过赋予初始值的方法声明变量类型：

```
var anInteger=0;
```

这将把 anInteger 定义为值为 0 的整型。

（3）变量的赋值

声明一个变量后，可以在任何时候对其赋值。赋值语句是：

<变量>=<表达式>;

其中“=”叫“赋值符”，它的作用是把右边的值赋给左边的变量。

2.7.3　表达式与运算符

表达式与数学中的定义相似，是指具有一定的值的用运算符把常数和变量连接起来的代数式。一个表达式可以只包含一个常数或一个变量。运算符可以是四则运算符、关系运算符、位运算符、逻辑运算符、复合运算符。下面来学习一些简单的表达式，见表 2-10。

表 2-10　基本表达式

表达式名称	表达式内容	表达式作用
括号	(x)[x]	中括号只用于指明数组的下标
自加	x++	x 值加 1，但仍返回原来的 x 值
	x--	x 值减 1，但仍返回原来的 x 值
自减	++x	x 值加 1，返回后来的 x 值
	--x	x 值减 1，返回后来的 x 值
乘、除	x*y	返回 x 乘以 y 的值
	x/y	返回 x 除以 y 的值
加、减	x+y	返回 x 加 y 的值
	x-y	返回 x 减 y 的值
关系运算	x<y x<=y x>=y x>y	当符合条件时返回 true 值，否则返回 false 值
等于、 不等于	x==y	当 x 等于 y 时返回 true 值，否则返回 false 值
	x!=y	当 x 不等于 y 时返回 true 值，否则返回 false 值
赋值、 复合运算	x=y	把 y 的值赋给 x，返回所赋的值
逻辑与	x&&y	当 x 和 y 同时为 true 时返回 true，否则返回 false
逻辑或	x||y	当 x 和 y 任意一个为 true 时返回 true，当两者同时为 false 时返回 false

2.7.4　JavaScript 中的语句

1. 注释语句

JavaScript 注释有两种：单行注释和多行注释。单行注释用双反斜杠“//”表示；多行注释是用“/*”和“*/”括起来的一行到多行文字。程序执行到“/*”处，将忽略以后的所有文字，直到出现“*/”为止。

如果程序需要草稿，或者需要让别人阅读，注释能帮上大忙。养成写注释的习惯，能节省你和其他程序员的宝贵时间，强烈推荐读者要养成合理书写注释的习惯。

2. 条件语句

在 JavaScript 中，条件语句有 if … else 和 switch 两种。

（1）if 语句

if 语句的格式为：

```
if(<条件>)<语句1> [ else <语句2> ];
```

若条件为 true，则执行语句 1；若为 false 则执行语句 2。

（2）switch 语句

switch 语句允许程序给表达式求值，并用 case 标记来匹配表达式可能的值，如果匹配成立，程序将执行相应的语句，用法如下：

```
switch(表达式){
case :语句1;
break;
case :语句2;
break;
…
default :语句n;}
```

switch 语句执行流程如下：

① 求表达式的值并依次序查看 case，直到找到一个匹配；
② 如果 case 的值等于表达式的值，则执行它相应的语句；
③ 继续执行，直到遇到一个 break 语句，或者 switch 语句结束。这意味着如果没有使用一个 break 语句，则多个 case 块被执行；
④ 如果没有 case 等于表达式的值，则跳转到 default。如果没有 default 情况，则跳转到最后一步；
⑤ 继续执行紧接 switch 代码块末尾的语句。

3. 循环语句

在 JavaScript 中，循环语句有 for 和 while 两种。

（1）for 语句

for 语句的格式为：

```
for(<变量>=<初始值>; <循环条件>; <变量累加方法>)<语句>;
```

下面看一个循环的例子，代码如下：

程序清单　文档 2-13.html

```
<html>
<head>
<title>for 循环</title>
 <script language="JavaScript">
 <!--
 var i;
 for(i=0;i<=5;i++){document.write("i="+i+"<br>");}
 //-->
 </script>
```

```
</head>
</html>
```

运行后的网页如图 2-12 所示。

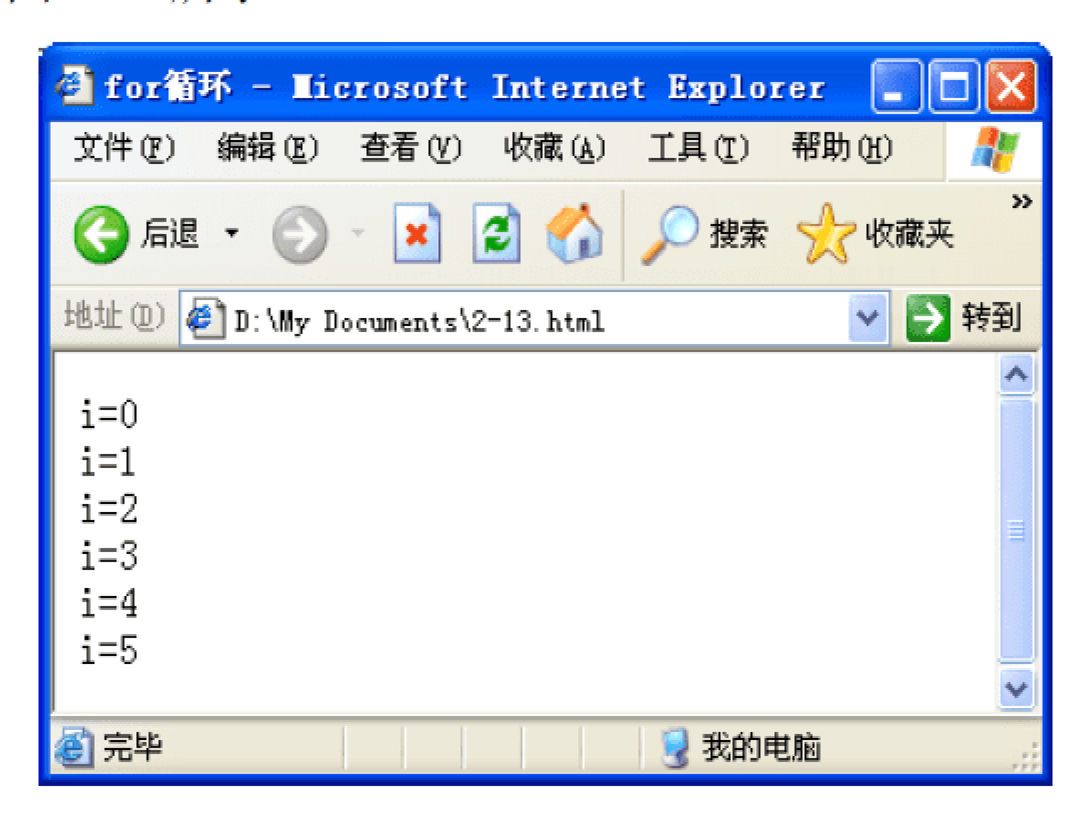


图 2-12 JavaScript 的 for 循环

（2）while 语句

while 语句的格式为：

while(<循环条件>)<语句>;

比 for 循环简单，while 循环是当满足<循环条件>时执行<语句>。while 循环的累加性质没有 for 循环强。

要想实现图 2-12 的网页，用 while 循环实现的 script 代码如下：

程序清单 文档 2-14.html

```
<script language="JavaScript">
<!--
var i=0;
while(i<=5){
document.write("i="+i+"<br>");
i++;}
//-->
</script>
```

以上对 JavaScript 的语法做了简单的介绍，实际上 JavaScript 的知识远远不止这些。关于 JavaScript 中对象和事件处理的有关知识，我们将在下一节的案例中直接边学边用。

2.8 案例 2 创建一个完整的用户注册页面

2.8.1 案例分析

本案例要求读者完成一个完整的用户注册页面，其中有对各项输入数据的检测，网站

上的注册页面和我们提供的范例有许多相似之处，可以仿照范例进行页面的合理验证。

2.8.2　实施步骤

1. 首先用 HTML 完成网页界面的设计。

本案例有 2 个文件，anli2.html 与 anli2.php。anli2.html 为用户填写注册信息的页面；anli2.php 为通过验证后显示的页面。

anli2.html 网页界面如图 2-13 所示。

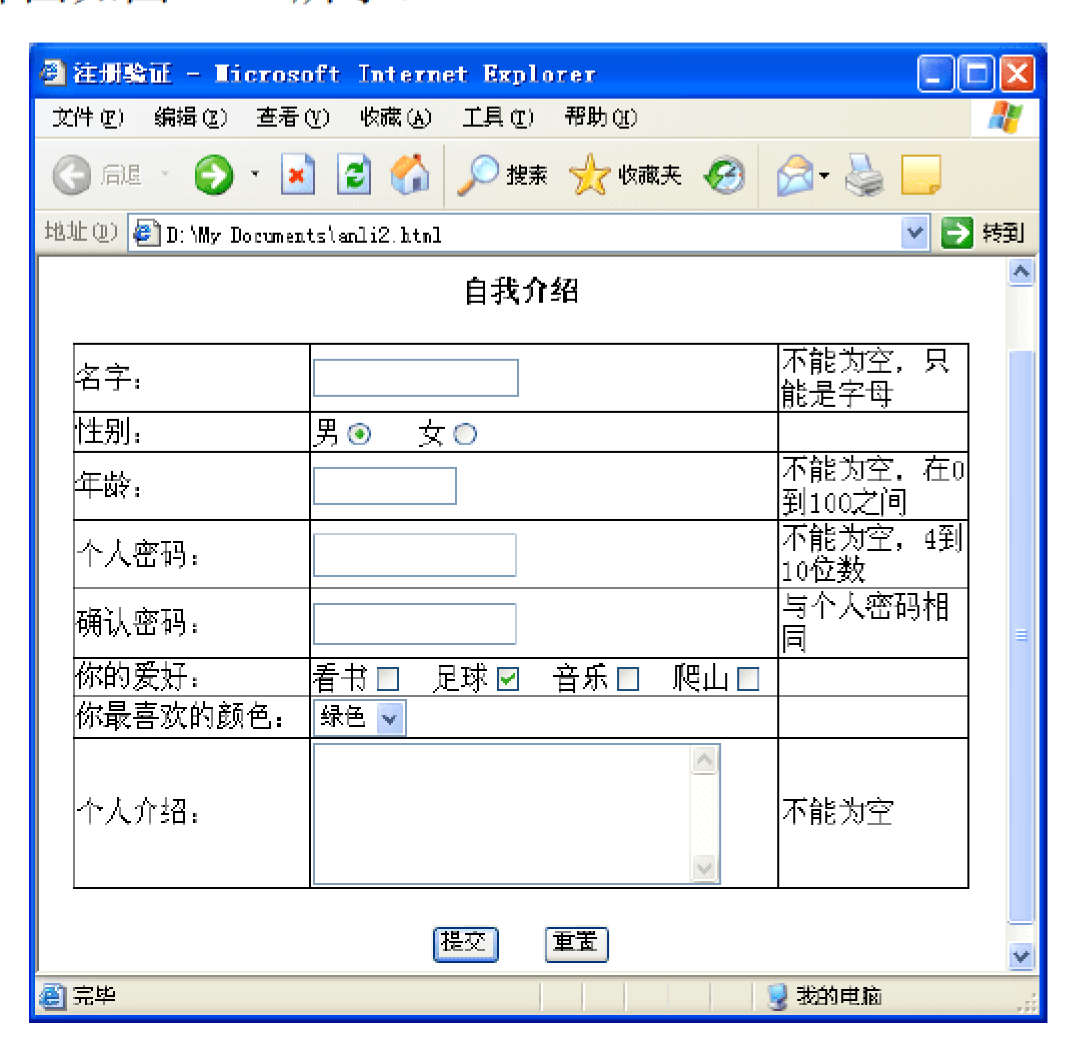


图 2-13　验证页面范例

anli2.php 用浏览器打开后的效果如图 2-14 所示。



图 2-14　成功通过验证

2. 添加 JavaScript 代码，用 JavaScript 完成对输入数据的检测。

3. 进一步的测试，使 JavaScript 代码起到关键的数据验证作用。

2.8.3　程序清单及源代码

1. 程序清单

（1）anli1.html

（2）anli1.php

2. 文件源代码如下

（1）anli1.html 源代码如下：

```
<html>
<head>
<title>注册验证</title>
<script language="JavaScript">
<!--
function namecheckdata(){
var name=document.myform.name,i,length=myform.name.value.length;
if(name.value==''){
  alert("用户名不能为空");
   name.focus();
 }
for(i=0;i<=length-1;i++){
if((name.value.charAt(i)>'a' &&
       name.value.charAt(i)<'z')||(name.value.charAt(i)>'A' &&
       name.value.charAt(i)<'Z')){} else
    {alert('用户名不符合规则');name.focus();name.select();}
  }
}//以上函数为名字合理性验证，用到 onblur 失去焦点事件。
function checkdata(){
var age=document.myform.age;
if(age.value==''){alert('年龄不能为空');age.focus();return false;}
if(!(age.value>0 && age.value<100)){alert
   ("你不是顽童就是寿星");age.focus();return false;}//验证年龄
if(myform.password.value.length<4||
   myform.password.value.length>10){alert('密码在 4 到 10 位之间');
   myform.password.focus();return false;}//验证密码位数
if(myform.password.value != myform.password1.value){alert
   ('密码不一致');return false;}//验证密码不一致
if(myform.jieshao.value==''){alert
   ('个人介绍不能为空');myform.jieshao.focus();return false;}
}
//-->
</script>
</head>
<body onload="myform.name.focus();">
<form action="anli2.php" method="POST" name="myform" onSubmit=
   "return checkdata();">
<p align="center"><b>自我介绍
```

```
</b></p>
<center>
<table id="table1" width="508" cellspacing="0" cellpadding="0"
  border="1" bordercolorlight="#000000" bordercolordark="#FFFFFF">
<tr><td height="22">名字：</td><td height="22"><input type=text
  name="name" size=15 maxlength=8 onBlur="namecheckdata();"></td>
<td height="22" width="105">不能为空，只能是字母</td></tr><tr>
<td height="22">性别：</td><td height="22">男<input type="radio"
  name="sex" value="男" checked>  
女<input type="radio" name="sex" value="女"></td><td height="22"
  width="105"> </td></tr><tr>
 <td height="24">年龄：</td><td height="24">
     <input type=text name="age" size="10" maxlength="8"></td>
 <td width="105" height="24">
     不能为空，在 0 到 100 之间</td></tr><tr><td>个人密码：</td><td>
     <input type=password name="password" size=15 maxlength=8></td>
 <td width="105">不能为空，4 到 10 位数</td></tr>
<tr><td>确认密码：</td><td><input type=password name="password1" size=
  15 maxlength=8></td>
<td width="105">与个人密码相同</td></tr><tr><td>你的爱好：</td><td>看书
  <input type="checkbox" name="like1" value="看书"> 
   足球<input type="checkbox" name="like2" value="足球" checked> 
   音乐<input type="checkbox" name="like3" value="音乐"> 
   爬山<input type="checkbox" name="like4" value="爬山"><!--多选按钮-->
</td><td width="105">  </td></tr><tr><td>你最喜欢的颜色：</td><td>
    <select name="color" size="1">
      <option name="red">红色
      <option name="green" selected>绿色
      <option name="blue">蓝色
   </select></td><td width="105">
      </td></tr><tr><td>个人介绍：</td><td>
<textarea cols="30" rows="5" name="jieshao"></textarea></td>
<td width="105">不能为空</td></tr></table>
<p></p>
<input type="submit" value="提交">  
<input type="reset">
</center>
</form>
</body>
</html>
```

（2）anli1.php 源代码如下：

```
<html>
```

```
<head>
<title>验证处理</title>
</head>
<body>
<center>恭喜你，您通过了验证！</center>
</body>
</html>
```

2.9 本章小结

本章中着重介绍了 HTML 源代码以及客户端 JavaScript 的应用，在 HTML 中，读者应掌握 HTML 文档结构、字体、表格、表单等的书写方法，在 JavaScript 中，应重点掌握 JavaScript 在表单验证中的作用。

2.10 练习题

1. 试写出 HTML 的字体、表格、表单的源代码。
2. 试写出 JavaScript 的条件语句、循环语句。
3. 用 JavaScript 完成对表单输入内容的验证，并实现注册网页的功能。

CHAPTER 3

PHP 服务器环境配置

本章重点

- PHP+IIS 的安装配置
- PHP+Apache 的安装配置

PHP + MySQL

3.1 PHP 服务器概述

有过网页制作或网站开发经历的读者应该知道 HTML 网页的基本运行原理，即客户端通过浏览器向服务器发出页面请求，服务器收到请求后直接将所请求的页面发回给客户端，然后客户端就能在浏览器中看到页面的显示效果。这是一个比较简单、直接的过程，只需要一台安装了 Web 服务软件的服务器就能完成。

PHP 和其他服务器端嵌入式脚本语言一样，需要首先搭建专门的服务器环境。只有配置好服务器环境，才能在服务器上运行 PHP 网站。PHP 网站和用其他语言开发的动态网站运行原理基本相同，其流程见图 3-1。

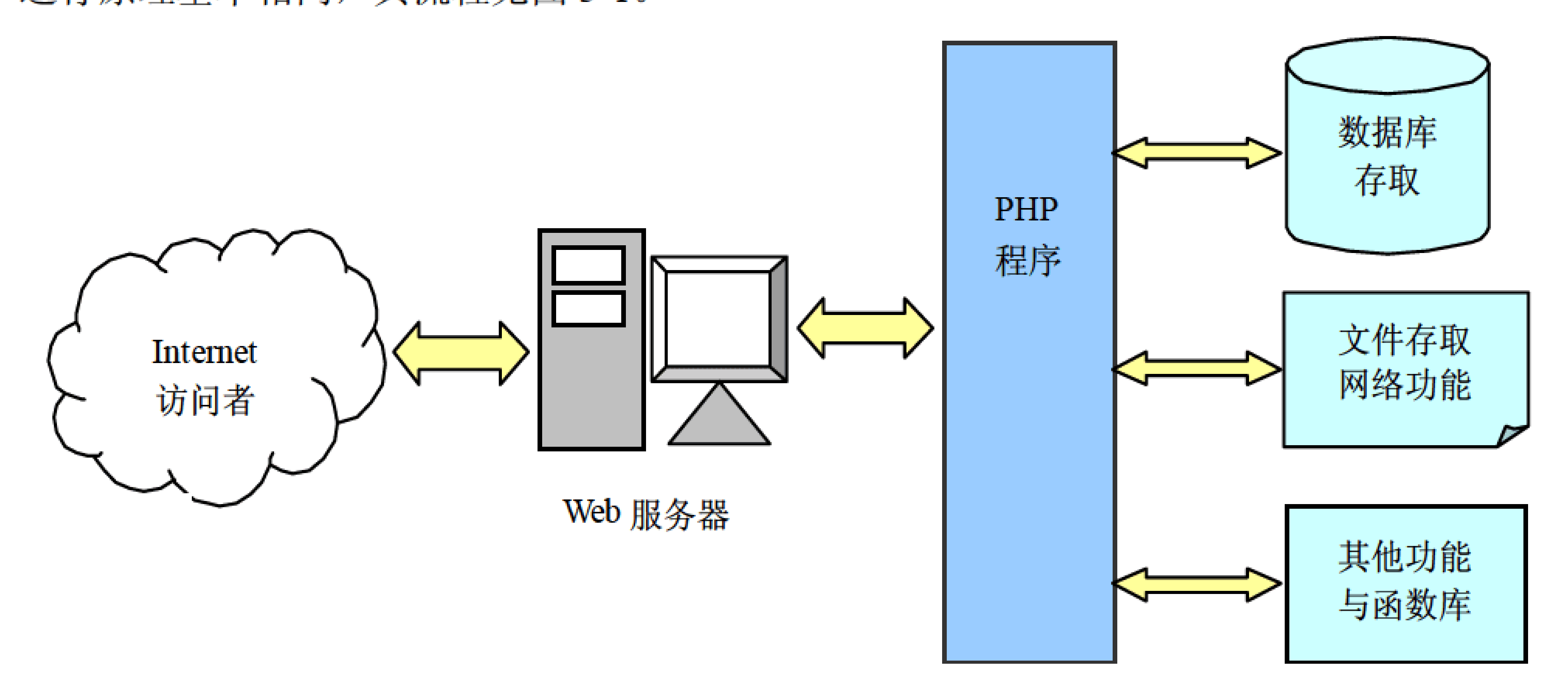


图 3-1　PHP 网站的运行流程图

通过图 3-1 可以看出，PHP 程序通过 Web 服务器接收访问请求，在服务器端处理请求然后再通过 Web 服务器向客户端发送处理结果。在客户端接收到的只是程序输出的处理结果，是一些 HTML 标记，而无法直接看到 PHP 代码，这样能够很好地保证代码的保密性和程序的安全性。此外，在服务器端运行代码还可以降低对客户端的要求，客户端不需要配置 PHP 环境，只需要安装普通浏览器即可浏览 PHP 网站。

PHP 是对服务器功能的扩展，服务器上安装了 PHP，就可以实现很多热门的应用。接下来就开始介绍 PHP 服务器环境配置的方法，让你的计算机也变成一台 PHP 服务器吧！

3.2 安装 PHP 前的准备工作

3.2.1 软硬件环境

安装 PHP 和安装其他软件一样，都需要硬件和软件环境。对于 PHP 来说，硬件的要求非常简单，在学习阶段只要有一台普通计算机就足够了。软件方面则需要根据自己的情况进行选择。这里主要从操作系统和 Web 服务软件两个方面来考虑。

PHP 能够运行在目前所有的主流操作系统上，包括 Linux、Unix 及其各种变种（包括 HP-UX、Solaris 和 OpenBSD）、Microsoft Windows 系列、Mac OS X、RISC OS 等。PHP

在这些平台上的安装步骤大同小异，就不一一介绍了。本书将以 Windows 平台为例，介绍 PHP 的安装和使用。由于 PHP 具有跨平台特性，所以在 PHP 开发阶段使用什么样的操作系统并不太重要，因为开发出来的程序可以很容易地移植到其他操作系统上去。

除了操作系统，另外和 PHP 的安装相关的就是 Web 服务软件，也称 Web 服务器。今天，PHP 已经支持了大多数的 Web 服务器，包括 Apache、Microsoft Internet Information Server（IIS）、Personal Web Server（PWS）、Netscape 以及 iPlant Server、Oreilly Website Pro Server、Caudium、Xitami、OmniHTTPd 等。这些 Web 服务器各有特点，目前以 APACHE 和 IIS 的使用最为广泛。本章中将会分别介绍 PHP 在 Apache 和 IIS 上的安装方法。

综上所述，要安装 PHP，首先必须保证计算机的操作系统和 Web 服务器已经安装并能够正常工作，然后就可以开始 PHP 的安装步骤。

3.2.2　获取 PHP 安装资源包

要安装 PHP，首先要获取 PHP 的安装资源包。这个资源包中包括了安装和配置 PHP 服务器的一切文件以及大量 PHP 扩展函数库。PHP 安装资源包的获取有很多途径，比如登录 PHP 官方网站下载，或者通过其他软件下载网站下载。因为 PHP 是免费软件，因此网上大量的软件下载站点都收录了此软件，可以通过搜索引擎搜索该软件，然后直接下载。

值得注意的是，很多网站提供的 PHP 安装资源包并不规范和全面。有些 PHP 安装软件是由第三方开发的，并没有经过 PHP 官方的认可和授权。有的安装包为了缩小体积还去掉了大部分扩展函数库，给以后的使用带来不便。因此，建议直接去 PHP 官方网站下载 PHP 安装资源包，并用手工方法安装和配置 PHP 服务器，只有这样才能保证所使用的 PHP 的完整、全面和权威。

现在就去 PHP 官方站点下载一个 PHP 安装资源包。登录 PHP 官方网站 www.php.net，如图 3-2 所示。



图 3-2　PHP 官方网站

单击导航链接的 Downloads 打开 PHP 的下载页面，该页面列出了最新的 PHP 安装资源包，如图 3-3 所示。单击 Windows Binaries 中的 PHP 5.1.6 zip package。

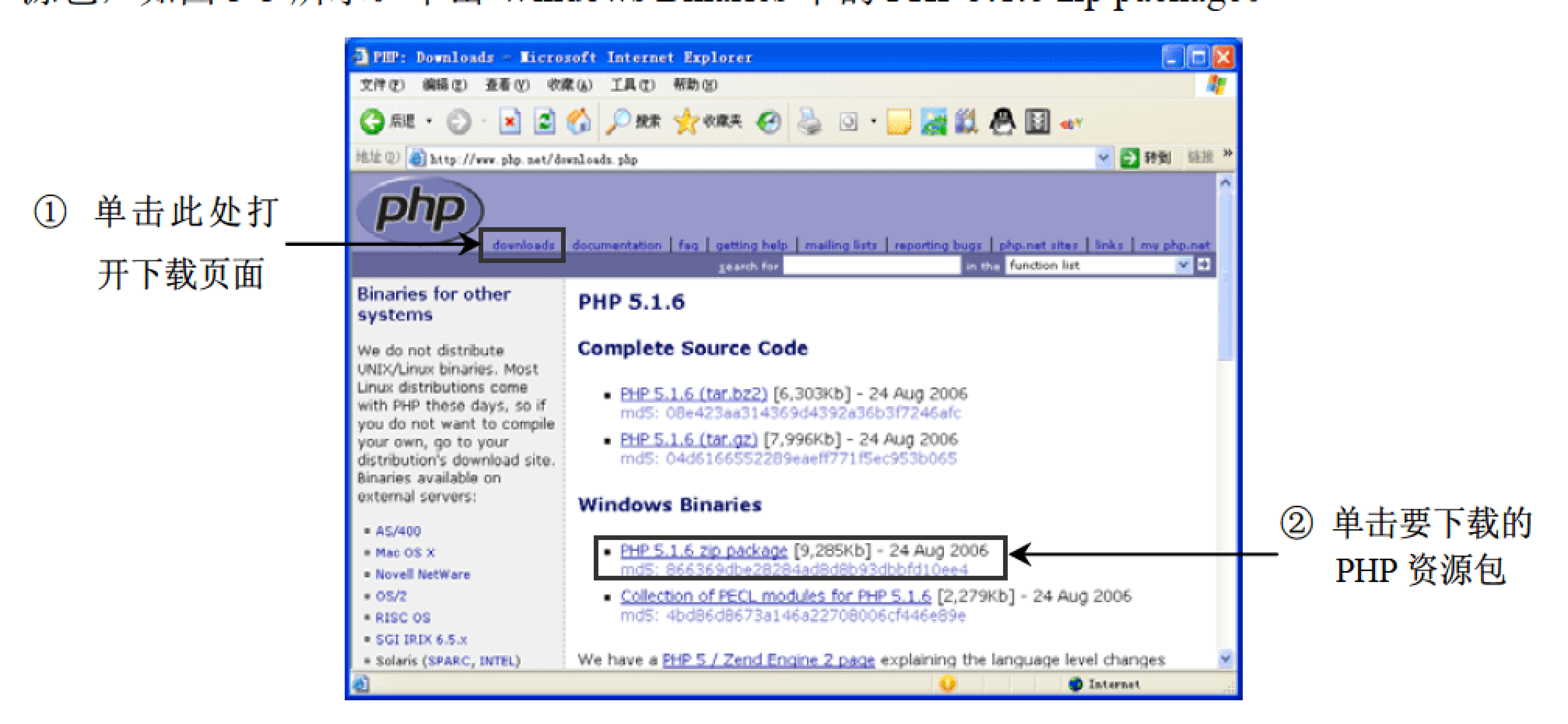


图 3-3 PHP 官方下载页面

单击此链接之后会出现一个镜像站点选择页面，通常网页会自动判断你的地域并给出推荐的镜像站点，直接根据推荐站点下载即可。如果推荐站点无法下载，可以选择一个离自己较近的镜像站点下载（见图 3-4）。

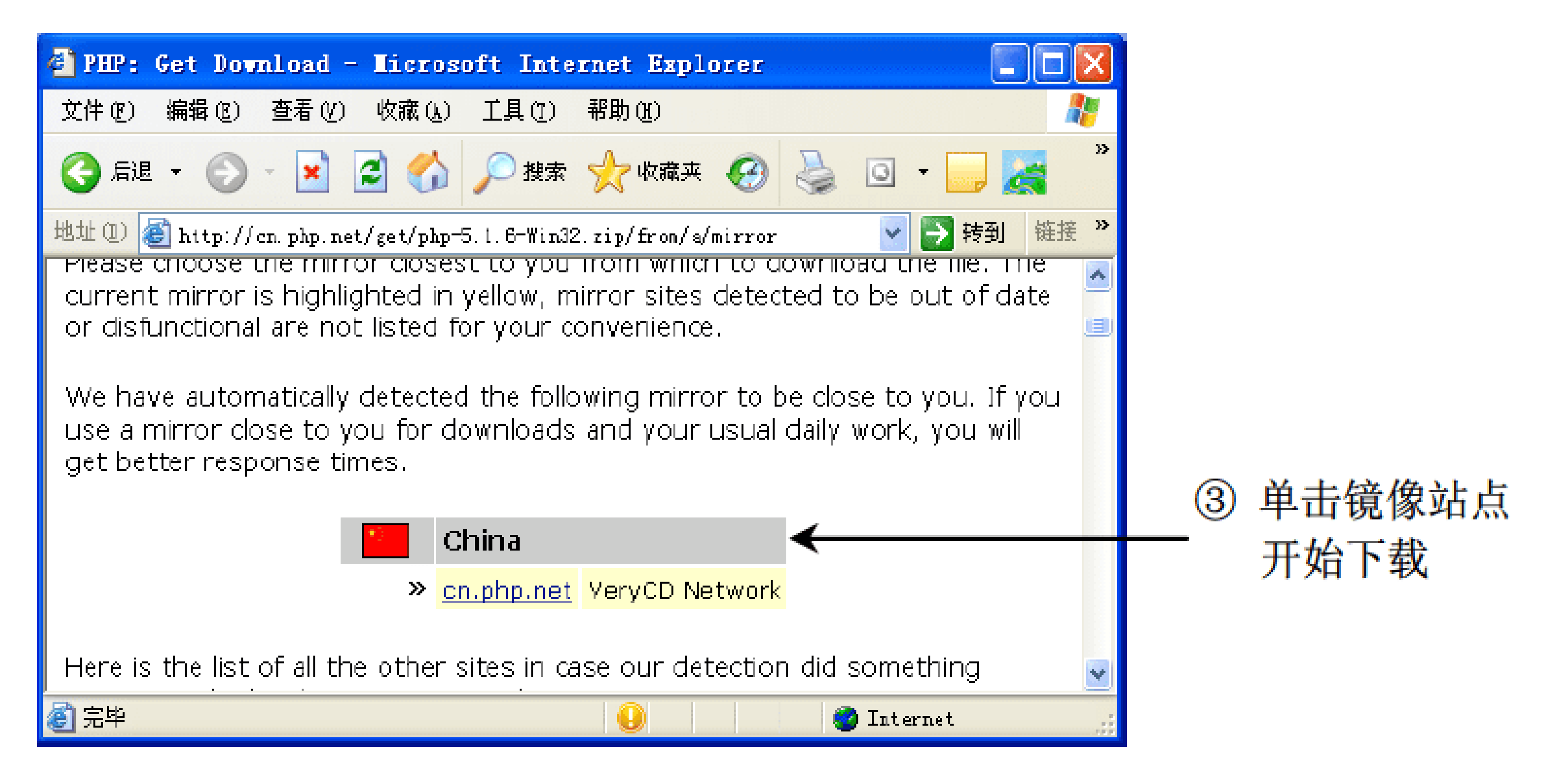


图 3-4 下载 PHP

下载完成之后，得到一个名为 php-5.1.6-Win32.zip 的压缩包。这就是需要 PHP 安装资源包。通过此压缩包的名字能看出其版本号为 5.1.6，适用平台为 Win32，即 Windows 系列。如果你下载到的资源包与此名字不符，应考虑是否单击了错误的链接。

到此为止，我们已经做好了 PHP 服务器搭建的准备工作，下面开始安装步骤了。

3.3 PHP 5+IIS 服务器安装配置

3.3.1 IIS 简介及其安装

IIS（Internet Information Server）是目前使用较为广泛的 Web 服务器之一。它操作简单，

使用方便，功能强大。IIS 由微软公司开发，目前也只能运行在微软公司的 Windows 系列操作系统上。包括 Windows 2000、Windows XP 及 Windows Server 2003 等版本。

部分版本的 Windows 操作系统如 Windows 2000 Server、Windows Server 2003 等，在安装系统时默认自动安装 IIS。其他版本的则不默认安装。要检查操作系统是否已经安装了 IIS，可以打开控制面板，找到“管理工具”（Windows XP 为“性能和维护”，然后找到“管理工具”），打开“管理工具”，看里面是否有“Internet 信息服务管理器”的快捷方式。如果有，则说明 IIS 已经安装，否则可能没有安装。另外一个快捷的方法是直接打开开始菜单的“运行”窗口，输入 inetmgr 命令，回车运行，看是否能打开 IIS 管理界面，如果能打开，则表示已经安装了 IIS，否则说明没有安装。

如果系统还没有安装 IIS，那么需要首先安装它。安装的方法很简单，打开控制面板，打开“添加或删除程序”，然后单击“添加/删除 Windows 组件”，稍等之后会出现“Windows 组件向导”界面，如图 3-5 所示。



图 3-5　Windows 组件向导

图中显示的是 Windows Server 2003 系统的“Windows 组件向导”界面。在其他版本的操作系统中可能会略有不同。在组件列表中选择“应用程序服务器”，双击打开，出现应用程序服务器对话框，在子组件列表中将“Internet 信息服务（IIS）”勾选，如图 3-6 所示。

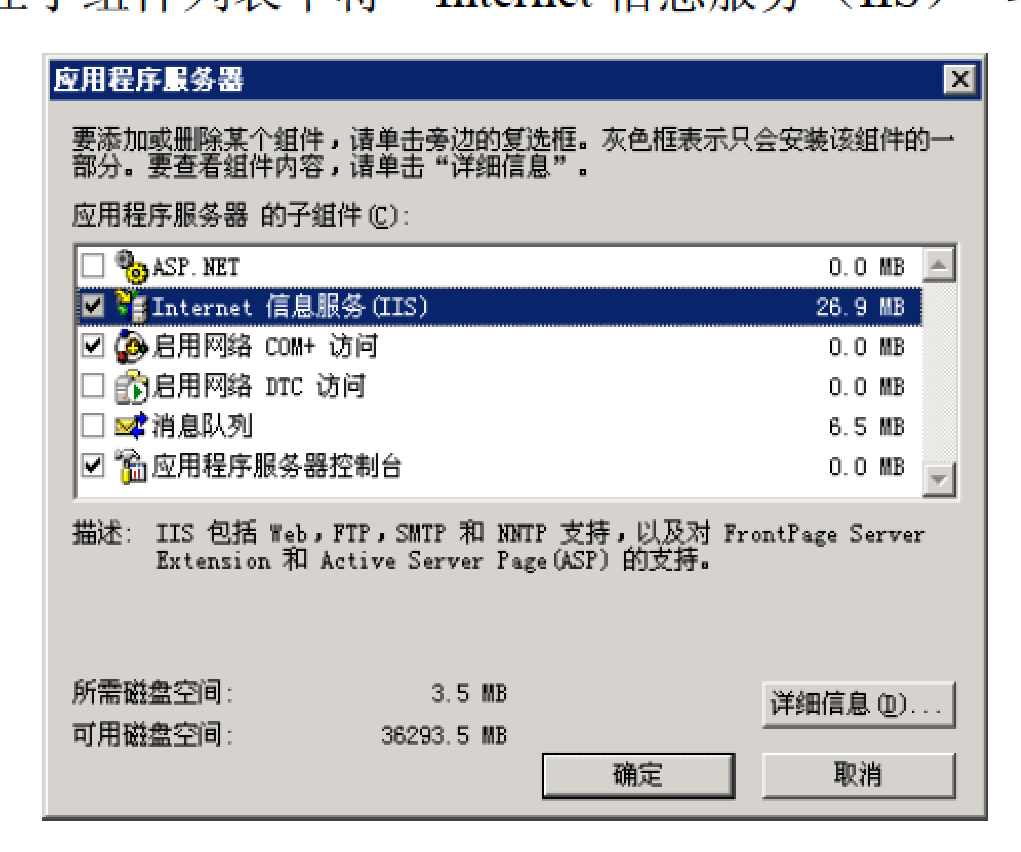


图 3-6　选择“Internet 信息服务（IIS）”

单击“确定”按钮，然后单击“下一步”，安装程序将开始运行。程序运行结束后，系统便安装好了 IIS。

在 Windows 2000/XP 系统中，打开“Windows 组件向导”界面后可以直接在列表中找到“Internet 信息服务（IIS）”，这时只要勾选此项然后单击“下一步”，即可完成安装。

3.3.2　PHP 的安装

系统安装 IIS 之后，下面可以开始 PHP 的安装了。本安装方法同样适用于 Windows 2000/XP/2003。

1. 解压和复制

在此之前我们已经下载了 PHP 安装资源包 php-5.1.6-Win32.zip，现在要做的第一件事就是将这个资源包解压缩，解压缩后会得到一个文件夹，这里面放着运行 PHP 所需要的一些文件和文件夹。接下来要将这些文件和文件夹拷贝到 PHP 的安装目录。PHP 的安装目录可以根据自己的情况随意设置。这里假设为 C:\php5\，如图 3-7 所示。

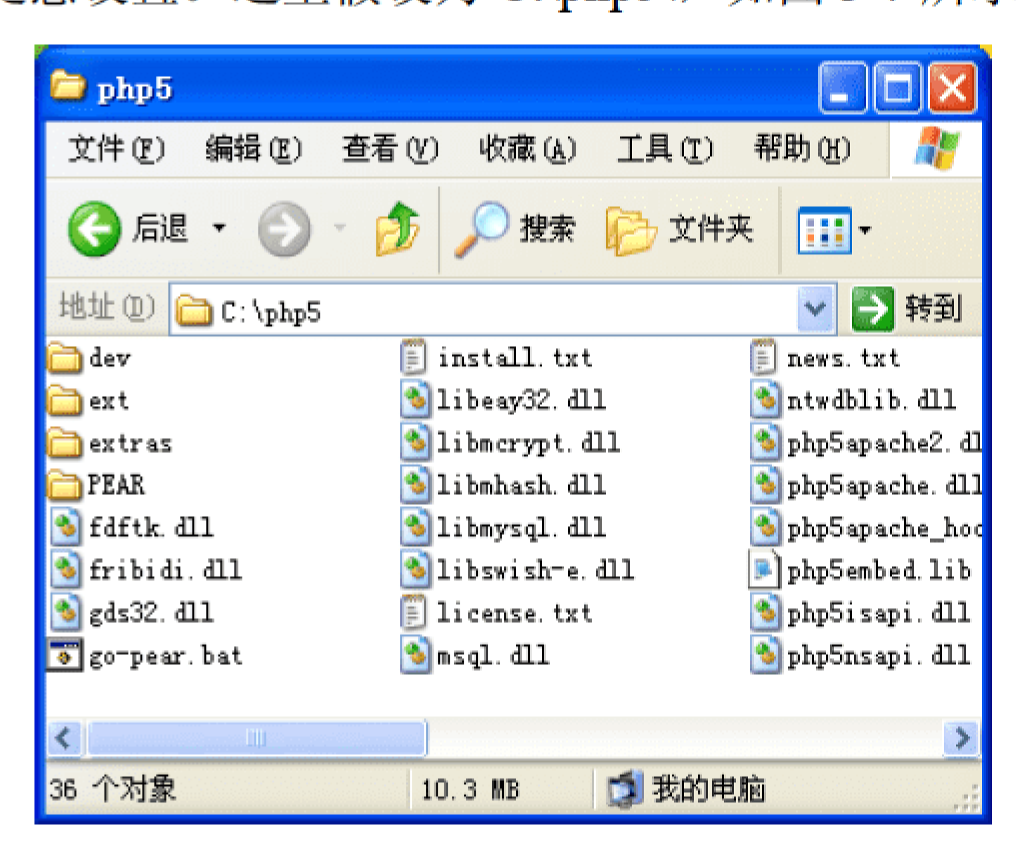


图 3-7　PHP 安装目录

2. 配置 php.ini

在安装目录下，找到一个名为 php.ini-dist 的文件，这个文件就是 PHP 的配置文件。这个文件里面存储了所有运行 PHP 时所需要的参数，通过修改这个文件可以实现对 PHP 的自由控制。现在打开并编辑这个文件，注意，现在这个文件的扩展名是.ini-dist，首先把扩展名修改为.ini，然后在记事本中打开。打开之后，可以看到许多令人眼花缭乱的参数，需要修改的参数十分有限。php.ini 中的参数形式为：

参数名=参数值

如 session.auto_start=0，前面的 session.auto_start 就是参数名，0 就是参数值。配置 PHP 参数，就是找到相关参数名，然后按照需要修改其值。

（1）**首先找到 short_open_tag 参数（可以用记事本的查找功能）**。这个参数用来设置是否可以使用“<?”短标记作为 php 代码开始的标记。意思是说，将来在 html 代码中嵌入 php 代码，将用什么标记开始。如果这个项目设置为 off，则只允许用“<?php”这种长标

记来作为 php 代码开始的标记。如果设置为 on，则可以使用“<?”或者“<?php”中的任意一种。

（2）**找到 register_globals 参数**。这个参数相当重要。很多 php 初学者或者曾经使用 PHP 3 版本的读者会在这个地方遇到麻烦。它用来设置是否允许 PHP 自动将外部提交的数据注册为全局变量，这是一个涉及到程序安全的参数。至于其如何影响程序安全，这里不再赘述，读者可以参阅相关资料，在这里建议将它设置为 off。

（3）**找到 extension_dir 参数**。此参数设置 PHP 扩展函数库的查找路径。将其值相应修改为“你的安装目录\ext\”。也就是图 3-7 中的“C:\php5\ext\”。

（4）**找到 file_uploads 参数**。还有下面连续的两个 upload_tmp_dir 和 upload_max_filesize。这三个参数是用来控制有关文件上传的。分别表示的意思是：是否允许文件上传（on/off）；上传文件的暂存路径；上传文件的最大字节数。可以根据需要来修改和填写，也可以使用默认值。upload_tmp_dir 最好填写，以保证将来可以顺利用 PHP 实现上传文件功能。可以在 C:\php5\下创建一个 UPLOADS 文件夹，并将 upload_tmp_dir 设置为“C:\php5\UPL- OADS\”。此处的文件夹可以任意指定，但必须保证指定的文件夹存在。

（5）**找到 session.save_path 参数**：这个参数用来设置 Session 的保存路径，在后面的章节中会详细讲解 Session 的使用。可以指定任意一个目录来存放 Session，在本例中，不妨设置为“C:\php5\sessions\”。当然也必须先创建这个目录。接着往下看几行，找到 session.auto_start 参数。对于初学者，建议将它设置为 1，这样服务器一启动，便自动支持 Session，在编写代码时就可以不必手工开启它。

到此为止，我们的 PHP 配置已经足够初学者使用很长时间了。当然 php.ini 中还有很多有趣和重要的配置，在这里无法一一提到。而且在没有深入理解这些配置的用途之前，还是采用默认值比较好。如果读者对此感兴趣，可以参考 PHP 手册以及 PHP 安装说明文件来了解更多的参数。但值得注意的是，对这个配置的修改必须足够慎重，因为这个配置直接关系到 PHP 的运行。

3. 添加系统变量

接下来要将 PHP 的安装目录添加到系统变量中去。这样就能保证 PHP 的配置信息和动态链接库能够被系统找到。当然你也可以把 php.ini 以及一些必要的链接库拷贝到现有的系统目录，但是那样做不利于 PHP 以后的升级和维护。将 PHP 的安装目录添加到系统变量是一个一劳永逸的办法。

在 Windows NT\2000\XP 和 2003 下，操作方法是：

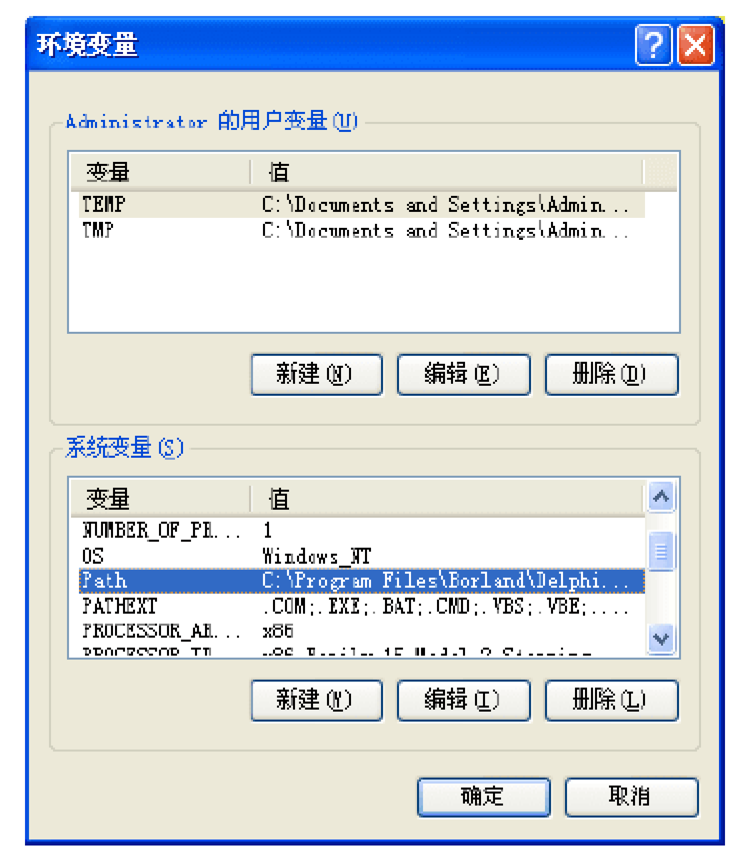
（1）进入控制面板并双击“系统”图标（或在桌面的“我的电脑”图标上单击鼠标右键，选择“属性”）。

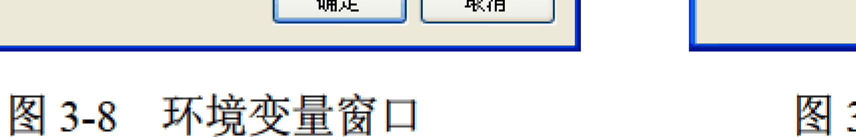
（2）选择“高级”标记页，单击“环境变量”按钮，打开环境变量设置窗口，如图 3-8 所示。

（3）在“系统变量”栏中找到 Path 这一项（有时需要需要拉动滚动条才能找到），选中此项并单击“编辑”按钮，出现变量修改对话框，如图 3-9 所示。在“变量值”一栏的后面将你的 PHP 安装目录添加上。本例中就是“;c:\php5”。注意不要遗漏了分号“;”，这

个分号的作用是与前面的一个路径分隔开来。

（4）依次单击“确定”按钮关闭窗口，然后重新启动计算机（必须重新启动，注销无效）。





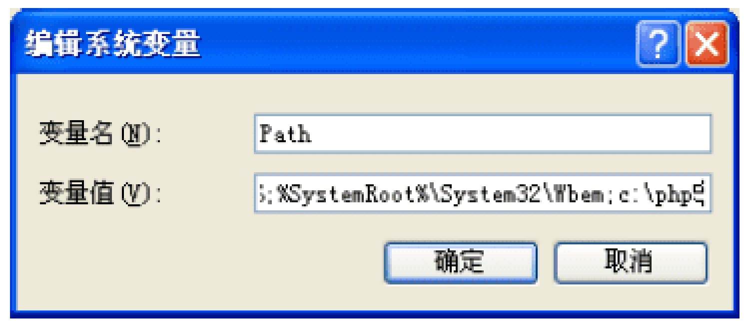


图 3-8 环境变量窗口 图 3-9 编辑系统环境变量

在 Windows 98/Me 操作系统上，需要修改 autoexec.bat 文件来添加环境变量。由于此操作系统目前已很少使用，在此不再详述其配置过程。

重新启动之后，PHP 便安装完成了。不过目前还无法编写程序来测试安装的效果，还需要将 PHP 与 IIS 相关联，服务器才算搭建完毕。

3.3.3 将 PHP 与 IIS 建立关联

只有 PHP 和 IIS 建立了关联，才能在 IIS 接收到对 PHP 的页面请求时，将处理权交给 PHP。下面以 IIS 5.1 版本为例，说明把 PHP 和 IIS 建立关联的方法。

（1）打开控制面板，找到“管理工具”中“Internet 信息服务”选项（Windows XP 系统为在控制面板的“性能和维护”中单击“管理工具”），或者直接在“开始”菜单中单击“运行”，打开“运行”窗口，在“运行”窗口中输入 inetmgr 命令，回车运行，打开 IIS 管理器，如图 3-10 所示。

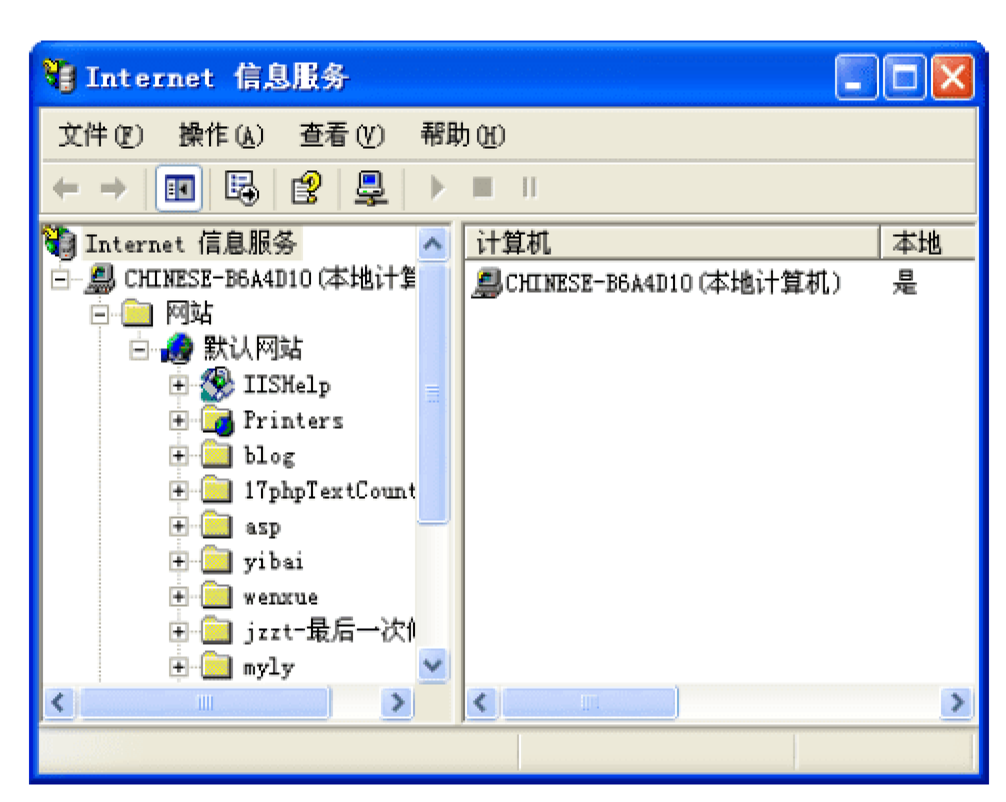


图 3-10 IIS 管理器

（2）在管理器左侧的目录树中找到“默认网站”，在上面单击鼠标右键，在菜单中选择“属性”，打开“默认站点属性”对话框。单击对话框上方选项卡中的“主目录”标记，切换到“主目录”选项卡，如图 3-11 所示。

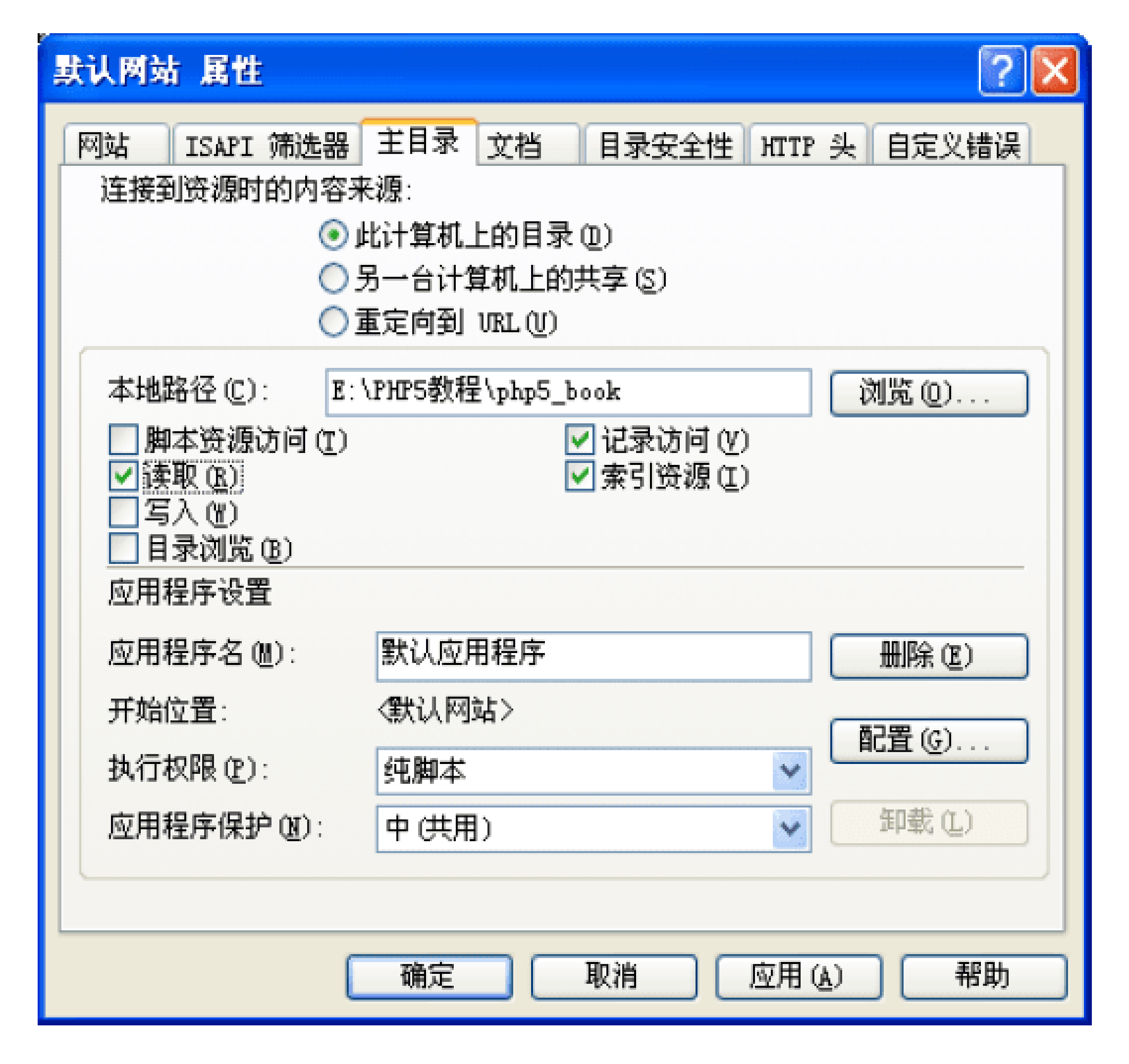


图 3-11　“默认站点属性”对话框

（3）然后单击面板下方的“配置”按钮，进入“应用程序配置”窗口，选择选项卡中的“映射”标记，打开“应用程序配置”面板，如图 3-12 所示。

（4）单击面板上的“添加”按钮，添加一个文件扩展名为.php 的映射。在打开的“添加/编辑应用程序扩展名映射”的窗口中，单击“可执行文件”中的“浏览”按钮，找到刚才的 PHP 安装路径“C:\php5\php5isapi.dll”（如果打开了这个目录后没有找到这个文件，那一定是在打开文件对话框时没有选择文件类型为“所有文件”或“*.dll 文件”），然后在下面的“扩展名”中填入 PHP 程序的扩展名，可以使用“.php”,“.php5”等等。注意下面的“脚本引擎”和“检查文件是否存在”前面要打勾选中（如图 3-13 所示）。

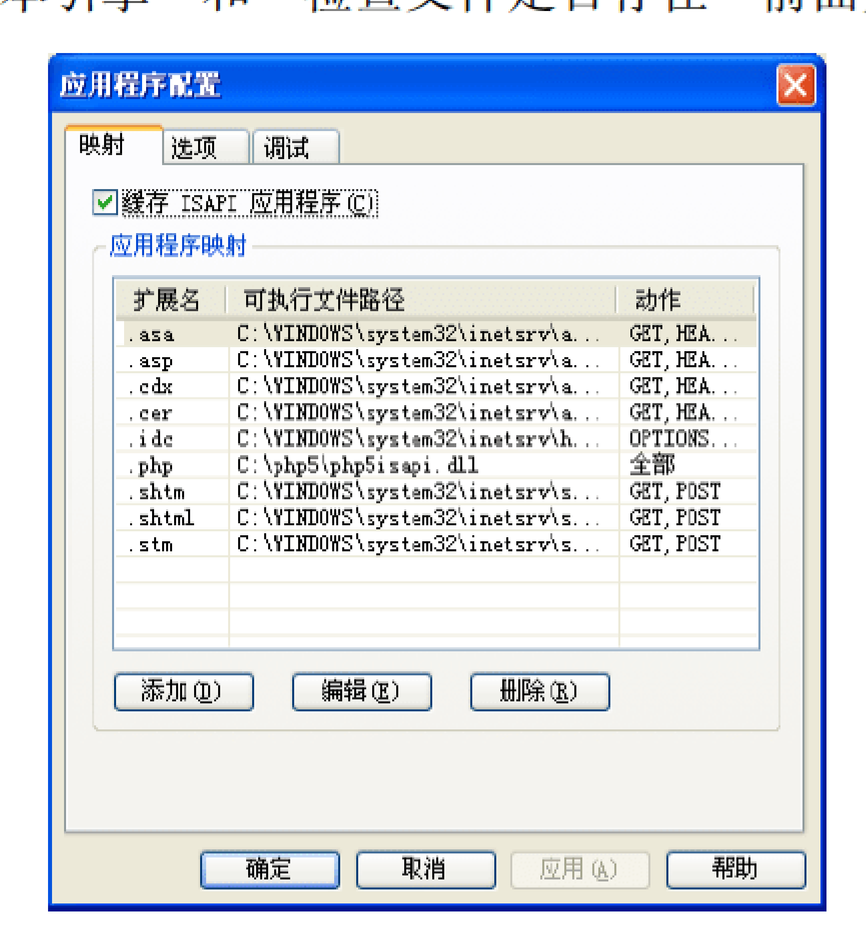


图 3-12　“应用程序配置”面板

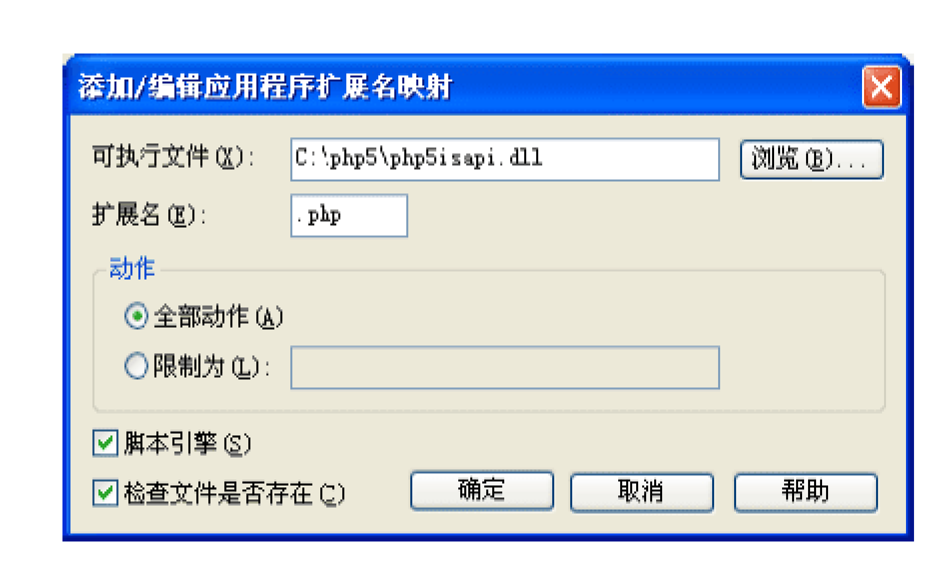


图 3-13　添加/编辑应用程序扩展名映射

（5）依次单击“确定”按钮关闭所有窗口，关闭 IIS 管理器，PHP 就完全配置完毕了。

您一定跃跃欲试，想立刻就编写一个 PHP 程序了吧，不要着急。现在需要完成最后一步操作——重新启动 IIS（如果觉得重新启动 IIS 比较麻烦，可以直接重新启动计算机）。在开始菜单中打开运行窗口，输入 net stop iisadmin 命令，回车运行。按照提示输入 Y，回车，稍等之后 IIS 就已停止。然后再在窗口中输入 net start w3svc 命令，回车运行，稍作等待之后 IIS 就已启动完成。至此，IIS 与 PHP 的关联工作大功告成！

如果你使用 Windows 2000 系统（IIS 5.0 版本），那么建立关联的方法和上面所讲的几乎完全相同，只有些许细微差别，读者可以自行试验一下。如果你使用 Windows Server 2003 系统（IIS 6.0 版本），那么你还需要进行一个步骤，就是添加应用程序扩展。方法如下：

（1）打开 IIS 管理器，在左侧目录树中找到“Web 服务扩展”选项，选择“添加一个新 Web 扩展”，如图 3-14 所示。



图 3-14　添加 Web 服务器扩展

（2）在打开的“新建 Web 服务扩展”对话框中，在“扩展名”中输入 php，在“要求的文件”中单击“添加”按钮，选择 PHP 安装目录下的 php5isapi.dll，并将下面的“设置扩展状态为允许”勾选，如图 3-15 所示。

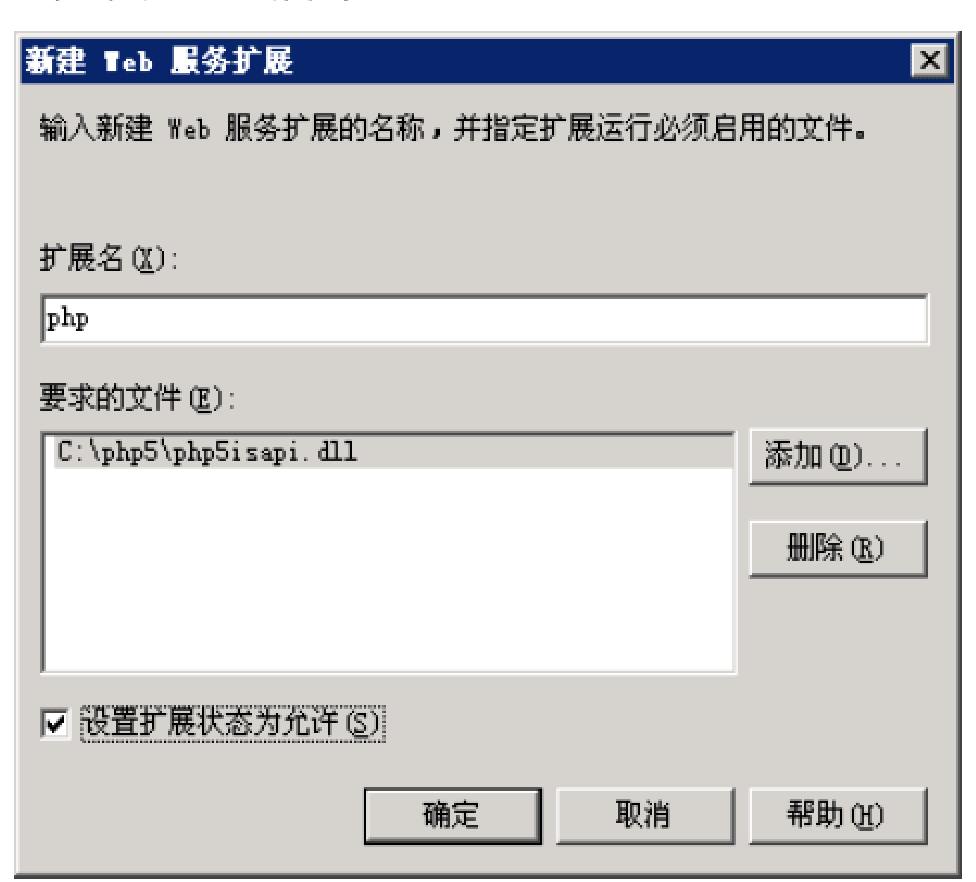


图 3-15　Web 服务扩展设置

（3）单击“确定”按钮，可以看到左侧的“Web 服务扩展”列表中已经有了名为 php 且状况为“允许”的服务扩展。至此，PHP 与 IIS 的关联工作全部完成!

3.3.4 设置主目录和虚拟目录

到此为止，PHP 环境搭建工作已经基本完成。但是在正式编写 PHP 程序之前，有必要先学习一下在 IIS 中如何设置主目录和虚拟目录。PHP 程序只有放在主目录或虚拟目录下，才能调试、运行。

所谓主目录，就是服务器的默认站点在服务器上的存放位置。比如某服务器 IP 地址为 10.0.0.10，那么当输入 http://10.0.0.10 这个地址来访问网站时，服务器如何知道网站存放的位置呢？假设网站存放在 D:\wwwroot\目录下，我们只要在 IIS 中将主目录设置为 D:\wwwroot\，那么在访问 http://10.0.0.10 这个地址时 IIS 会自动到此目录下搜索文件。比如我们写好了程序 index.php，将其存放在 D:\wwwroot\，这时只需要输入 http://10.0.0.10/index.php 即可看到该程序的运行结果。

在 PHP 学习阶段，一般都是以本地机器作为服务器，因此一般采用 http://127.0.0.1 或者 http://localhost 这样的地址来访问。本书中的例子都是在本地服务器的主目录或者虚拟目录下运行的。

1. IIS 服务器主目录设置

以 IIS 5.1 为例介绍，其他版本的 IIS 也大同小异。设置步骤如下：

（1）打开 IIS 管理器，在左侧树形菜单中选择展开“网站”，在“默认网站”上单击右键，选择“属性”，如图 3-16 所示。

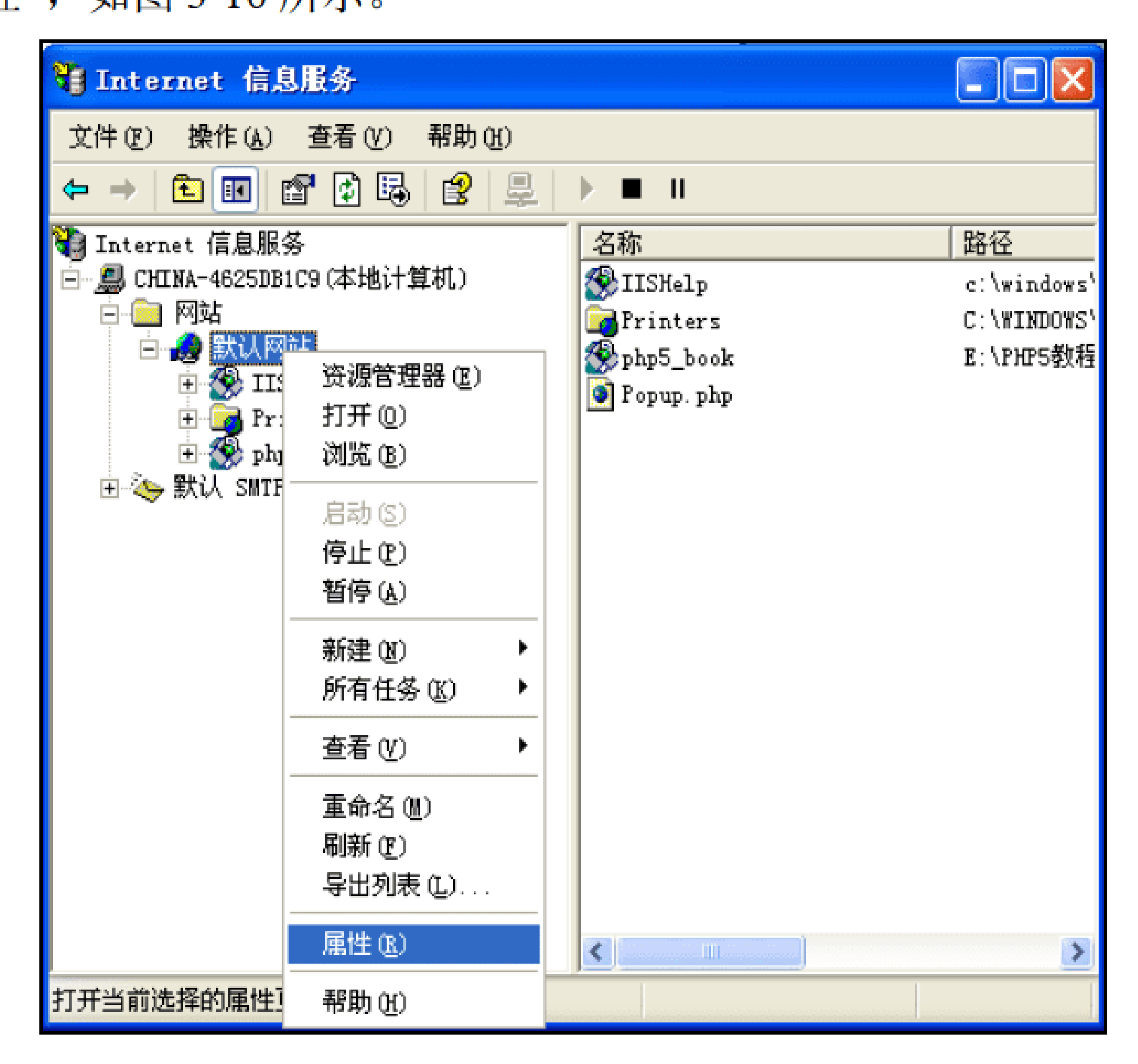


图 3-16 设置主目录（1）

（2）单击“属性”之后弹出“默认网站属性”对话框。在上面的标记中选择“主目录”，

打开主目录管理选项卡，如图 3-17 所示。

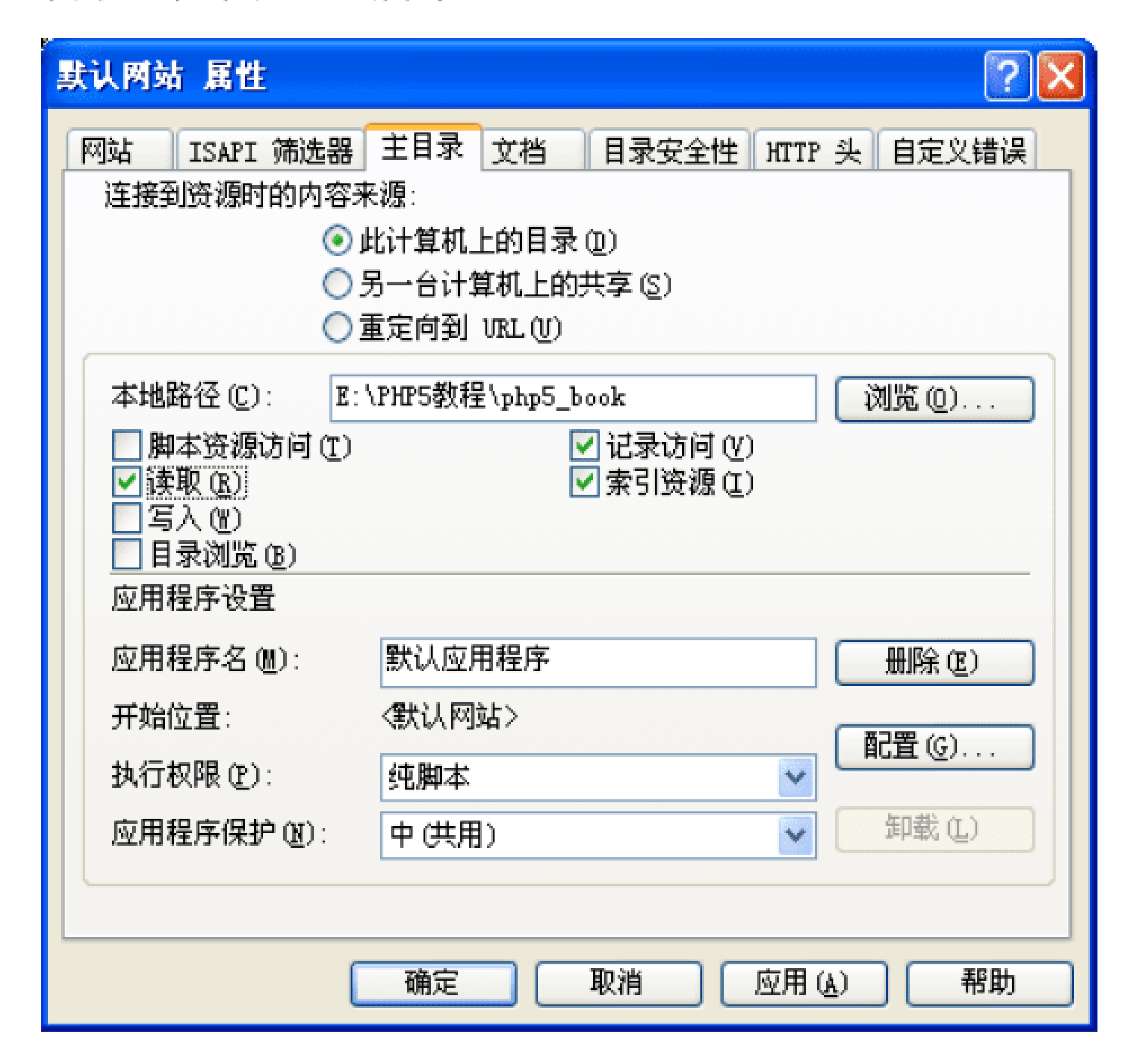


图 3-17　设置主目录（2）

（3）在主目录选项卡中，单击“浏览”按钮，选择计算机上的一个目录作为主目录。选择好之后依次单击“确定”按钮关闭所有对话框。

这样，IIS 的主目录就设置完成。使用“http://localhost/文件名”这样的地址就可以访问此目录下的文件。如果主目录下还有子目录，那可以用“http://localhost/子目录名/文件名”来访问。

2. IIS 虚拟目录设置

有时候我们希望把一些网页单独存放在计算机上的某个目录下，而不是全部集中存放在网站主目录下，这时候可以采用虚拟目录。虚拟目录和主目录不同，一台服务器只能有一个主目录，但可以有多个虚拟目录。虚拟目录访问的方式是在服务器地址后面加一个虚拟目录名。如“http://localhost/虚拟目录名/”。

虚拟目录之所以称之为“虚拟”，还有另外一层含义。那就是通过地址栏里的访问地址，无法确定网页的存放位置。比如“http://localhost/bbs/”这个地址，无法确定“/bbs”这个目录是主目录下面的一个子目录还是一个虚拟目录。因为虚拟目录可以存在于服务器上的任何位置，因此这就隐藏了实际的网页存储位置。这对于网站安全有所帮助。

创建一个虚拟目录的步骤如下：

（1）打开 IIS 管理器，展开“网站”，在“默认网站”上单击右键，选择“新建”选择“虚拟目录”菜单。打开“新建虚拟目录向导”对话框。如图 3-18 所示。

（2）单击“下一步”按钮，在“别名”输入框中输入该虚拟目录的名字（虚拟目录名任意，不必和文件夹名相同），假设取名为“ceshi”（见图 3-19）。



图 3-18　新建虚拟目录（1）

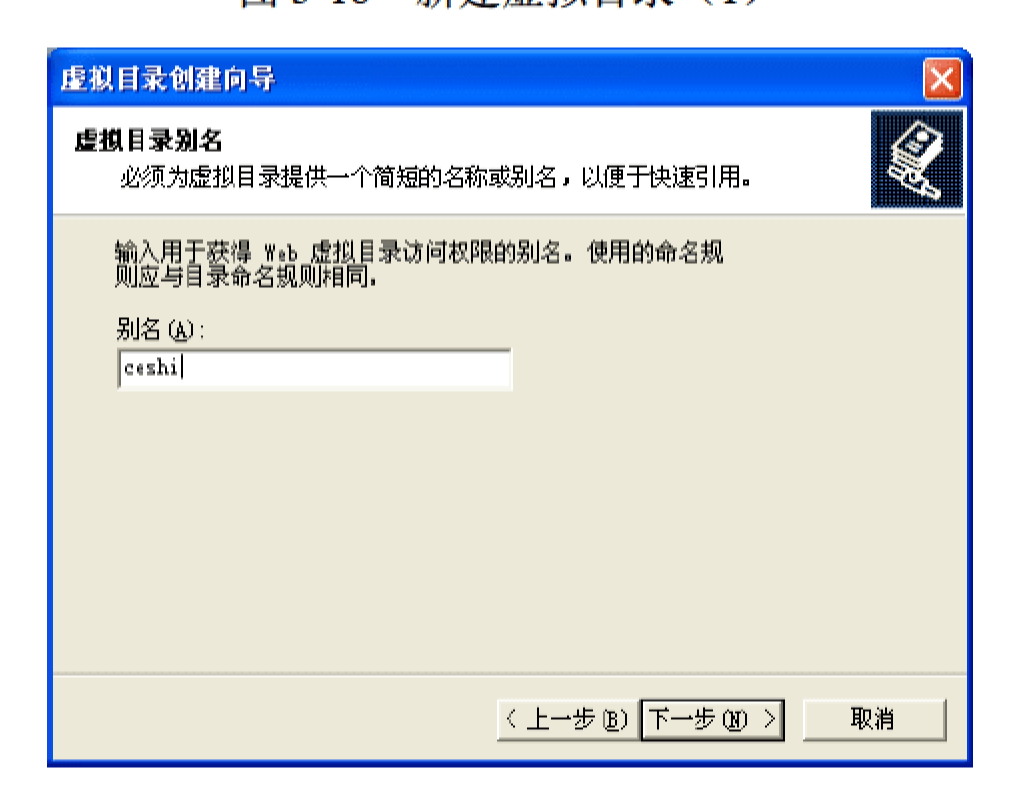


图 3-19　新建虚拟目录（2）

（3）单击“下一步”按钮，出现网站目录选择对话框。单击对话框后面的“浏览”按钮，选择一个目录作为该虚拟目录要访问的位置，也就是文件的存放位置。这里假设选择 D:\wwwroot\。单击“下一步”，出现“访问权限”对话框，要求选择该虚拟目录允许进行的操作。对于一般的 PHP 站点，只选择前三项即可。即“读取”、“运行脚本（如 ASP）”和“执行（如 ISAPI 应用程序或 API）”。如图 3-20 所示。

（4）单击“下一步”按钮，单击“完成”按钮，即可完成虚拟目录的创建工作。创建完成之后会在左侧的树形菜单中出现虚拟目录名。如图 3-21 所示。

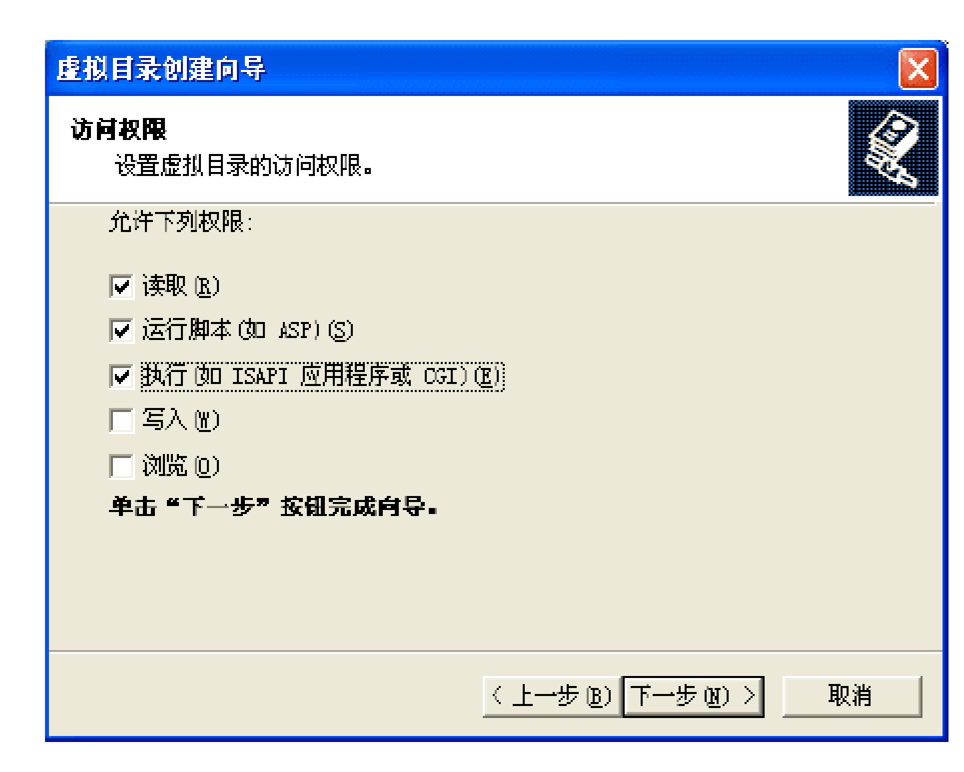


图 3-20　新建虚拟目录（3）

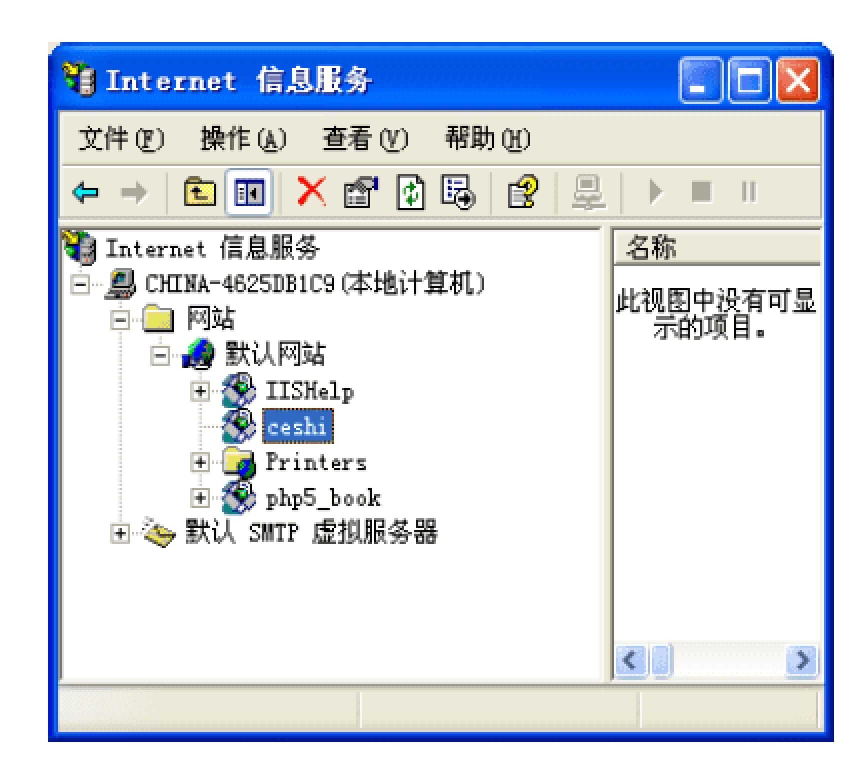


图 3-21　新建虚拟目录（4）

虚拟目录创建完成后，不论是 HTML 静态网页还是 PHP 程序，都可以放在 D:\wwwroot\ 目录下，然后用“http://localhost/ceshi/文件名”这样的地址访问即可！

另外，虚拟目录创建完成后，还可以对其属性进行设置和修改。方法与设置主目录相同，只需要在相应的虚拟目录的名称上单击右键，选择“属性”，在打开的对话框中进行操作即可。

上面简要介绍了在 IIS 5.1 中设置主目录和虚拟目录的方法。在其他版本的 IIS 上设置的方法基本雷同，读者可以自行探索、试验。读者应灵活掌握 IIS 的基本操作和设置方法，才能更好的学习 PHP 编程和网站开发。

3.3.5 编写并运行一个 PHP 测试程序

现在，你完全可以编写一个 PHP 程序了！是不是早已经等不及了呢？使用任意文本编辑软件（如 Windows 的记事本、UltraEdit 等）新建一个文本文件，键入下列代码：

程序清单　文档 3-1.php

```
<!--文件 3-1.php：PHP 测试程序-->
<HTML>
<HEAD>
<TITLE>PHP 环境信息</TITLE>
</HEAD>
<BODY>
  <?php
    phpinfo();
    ?>
</BODY>
</HTML>
```

注意，此程序不能像 HTML 网页那样直接用浏览器打开来看运行效果，PHP 程序写好之后首先需要由服务器来解释，服务器将 PHP 程序的运行结果以 HTML 格式返回给浏览器，才能看到效果。具体地说，就是首先要在服务器上建立一个主目录或者虚拟目录，然后将这个文件保存到你的主目录或者某个虚拟目录下。在本例中，网站主目录为“E:\PHP 5 教程\php5_book\”，因此可以将 3-1.php 存放到此目录下，然后打开浏览器，在地址栏中输入 http://localhost/3-1.php 或 http://127.0.0.1/3-1.php 然后回车。如果看到了如图 3-22 所示的 PHP 环境信息的页面，那么 PHP 服务器环境已经配置成功了！

也可以把 3-1.php 放在创建好的虚拟目录下。还是在本例中，将程序拷贝到 D:\wwwroot\ 下，然后在浏览器地址栏中输入 http://localhost/ceshi/3-1.php，也可以看到此程序的运行结果。

值得注意的是，采用了 NTFS 文件系统的服务器，有时会因文件访问权限问题导致 PHP 无法正常运行。因此，如果服务器采用了 NTFS 文件系统，那么至少应保证 php.ini 和 PHP 安装目录以及 PHP 程序所在的目录都拥有足够的访问权限（推荐为 Everyone 可访问）。

到此为止，已经将 PHP+IIS 服务器环境配置讲完了。虽然不同版本的 IIS 在配置时存在微小差异，但基本的设置都是相同的。读者可以自行实践一下在各个不同版本的 IIS 上配置 PHP 的方法。

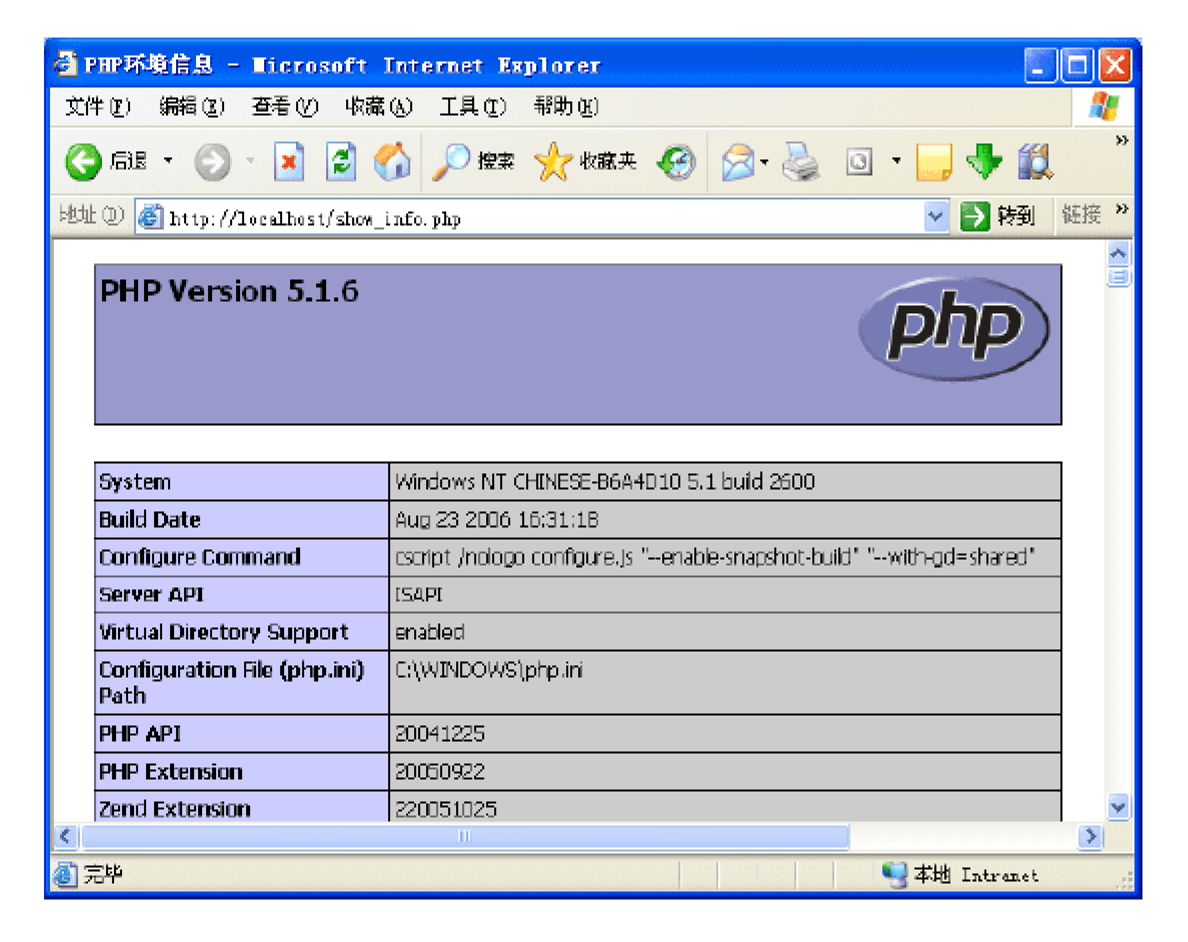


图 3-22　PHP 环境信息测试

3.4　PHP 5+Apache 服务器安装配置

3.4.1　Apache 简介

Apache 是目前世界上使用最为广泛的 Web 服务器之一，根据 NetCraft 所作的调查，世界上 50%以上的 Web 服务器都在使用 Apache。

1995 年 4 月，最早的 Apache 0.6.2 版由 Apache Group 公布发行。Apache Group 是一个完全通过 Internet 进行运作的非营利机构，由它来决定 APACHE Web 服务器的标准发行版中应该包含哪些内容。Apache 为开放源代码软件，允许任何人对其修改、扩充、更新。和其他服务器相比，Apache 拥有以下主要特性：

（1）几乎可以运行在所有的计算机平台上；

（2）支持最新的 http/1.1 协议；

（3）简单而且强有力的基于文件的配置（httpd.conf）；

（4）支持通用网关接口（CGI）、FASTCGI。支持虚拟主机，支持 http 认证；

（5）具有对用户会话过程的跟踪能力；

（6）支持 Java Servlets；

（7）运行效率高，成本低。

当前，Apache 已经发展到 2.2.3 版本。不过常用的 Apache 版本有 APACHE 1.3 和 Apache 2.0.x。由于 PHP 和 Apache 的不同版本之间存在兼容性问题，因此在选择 Apache 的版本的同时应当考虑 PHP 的版本。在本书中使用的 PHP 为当前最新的 PHP 5.1.6，所以可以使用 Apache 2.0.x 版本与之搭配。

3.4.2　Apache 的获取

由于 Apache 是免费软件，因此有很多途径可以获取 Apache 的安装文件。最稳妥的方法当然还是从其官方网站直接下载。Apache 的官方网站是 http://www.apache.org。

打开官方站点首页，在左侧的产品分类中选择 HTTP Server，在打开的网页中选择合适的 APACHE 版本，单击 Download，即可进入下载页面。单击相应下载链接即可获得 Apache 安装文件，如图 3-23（a）所示。

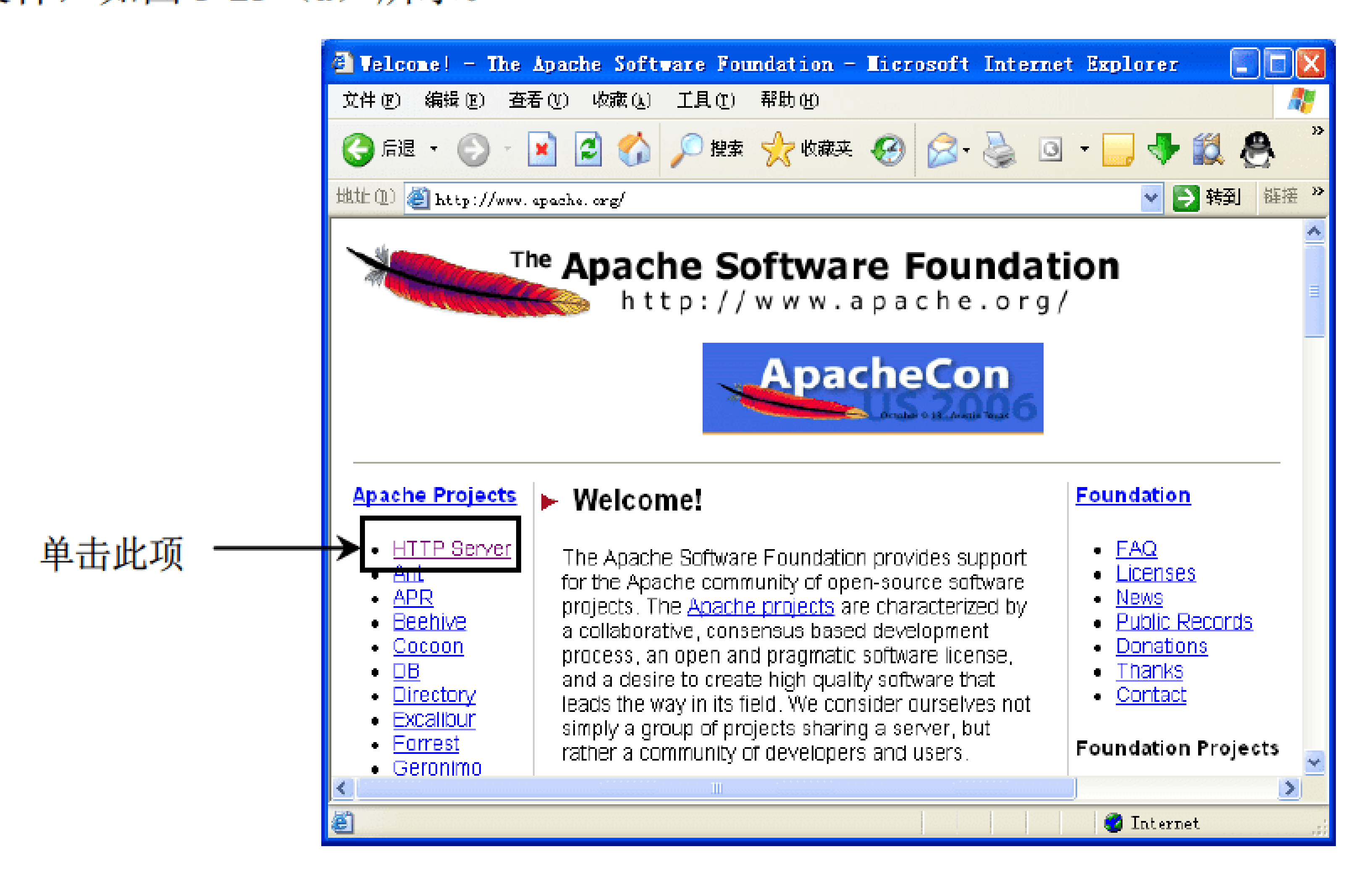


图 3-23（a）Apache 官方网站

在这里，我们下载的是 Apache 2.0.59，文件名为 apache_2.0.59-win32-x86-no_ssl.msi，这是一个可以直接双击运行的安装程序。它属于 Apache 2.0.x 系列，能够运行于 Windows 系统平台，并与 PHP 5 搭配使用。如图 3-23（b）所示：

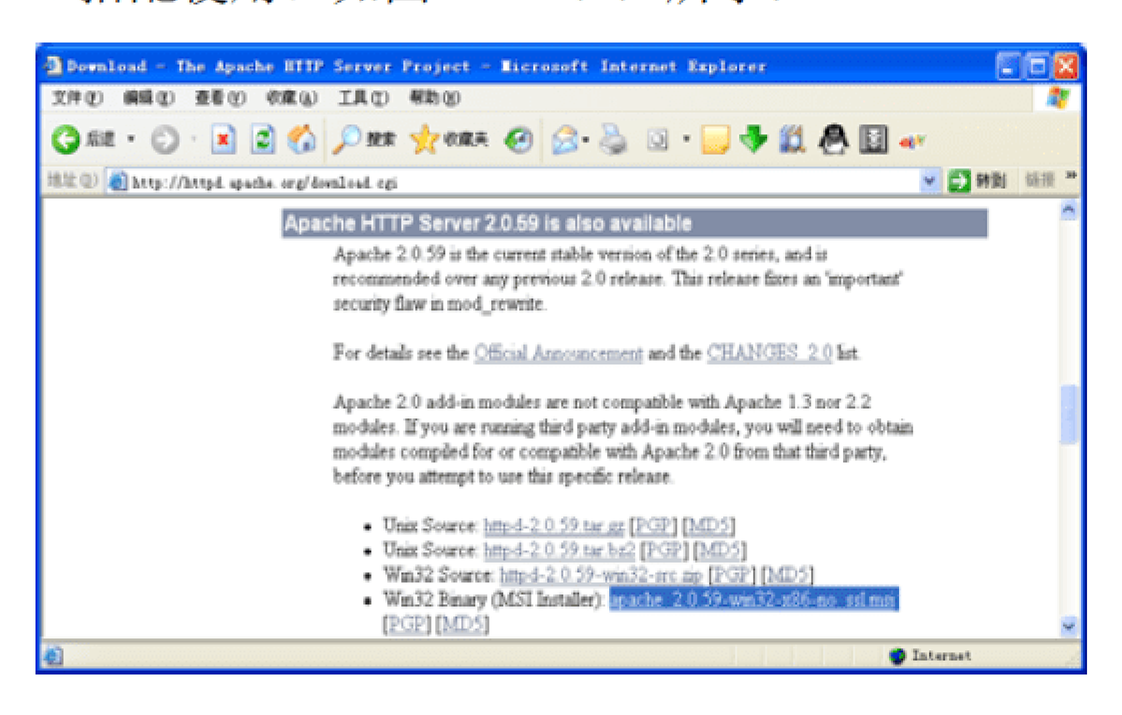


图 3-23（b） 下载 Apache

3.4.3 Apache 的安装

Apache 在 Windows 平台上的安装非常简单，只需要启动安装程序，按照提示一步步进行即可。不过在安装 Apache 之前，必须保证你的机器没有安装 IIS 或其他服务器软件，如果已经安装了，建议将其卸载。因为可能会由于端口冲突而导致 Apache 无法启动。当然你可以通过一些设置使各个服务器软件工作在不同端口，但是在绝大多数情况下只需要一个服务器软件就足够了。

双击启动 Apache 安装程序，出现软件的欢迎界面，单击 Next 按钮，出现 Apache 许可协议，阅读完毕许可协议之后选择 I accept the terms in the license agreement，表示接受许可

协议中的条款，再单击 Next 按钮，出现 Apache 服务器注意事项，阅读完毕后单击 Next 按钮，出现服务器信息设置界面。如图 3-24 所示。此处要求设置一些服务器基本信息。分别是网络域名、服务器名、管理员信箱以及 Apache 的工作方式。如果只是在自己的机器上使用 Apache，前三个选项可以保持空白，不需要设置。第四个选项有两种选择，建议选择第一项，也就是“针对所有用户，工作在 80 端口，安装为服务”。“安装为服务”的意思是将 Apache 安装为 Windows 的一个服务，当机器启动时自动启动 Apache。

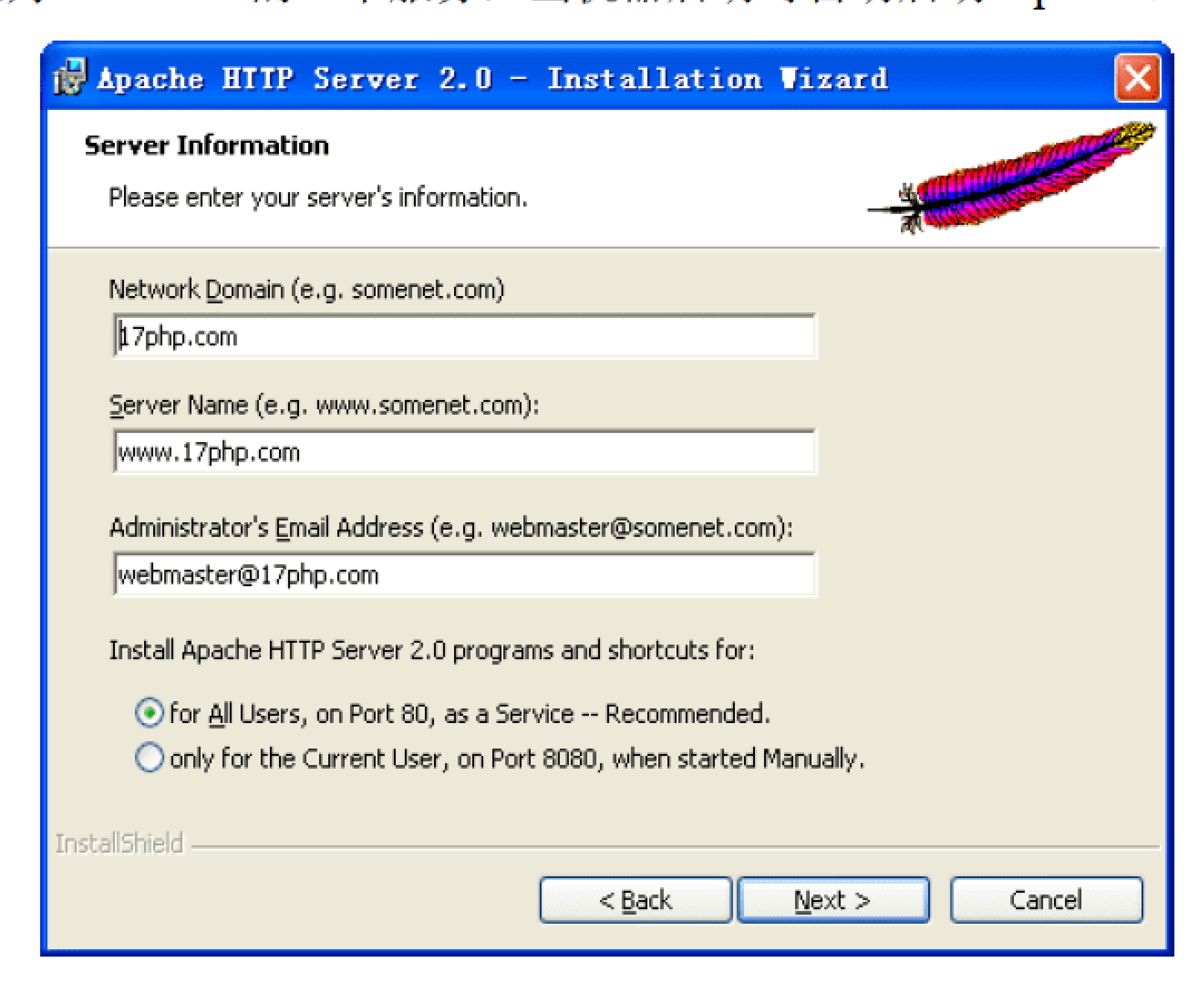


图 3-24　Apache 服务器信息设置

设置完成之后，单击 Next 按钮进入下一步。这时出现安装类型窗口，在这里有 Typical 和 Custom 即“典型安装”和“用户自定义安装”两种选择，对于对不太熟悉的初学者，建议使用“典型安装”。如图 3-25 所示。

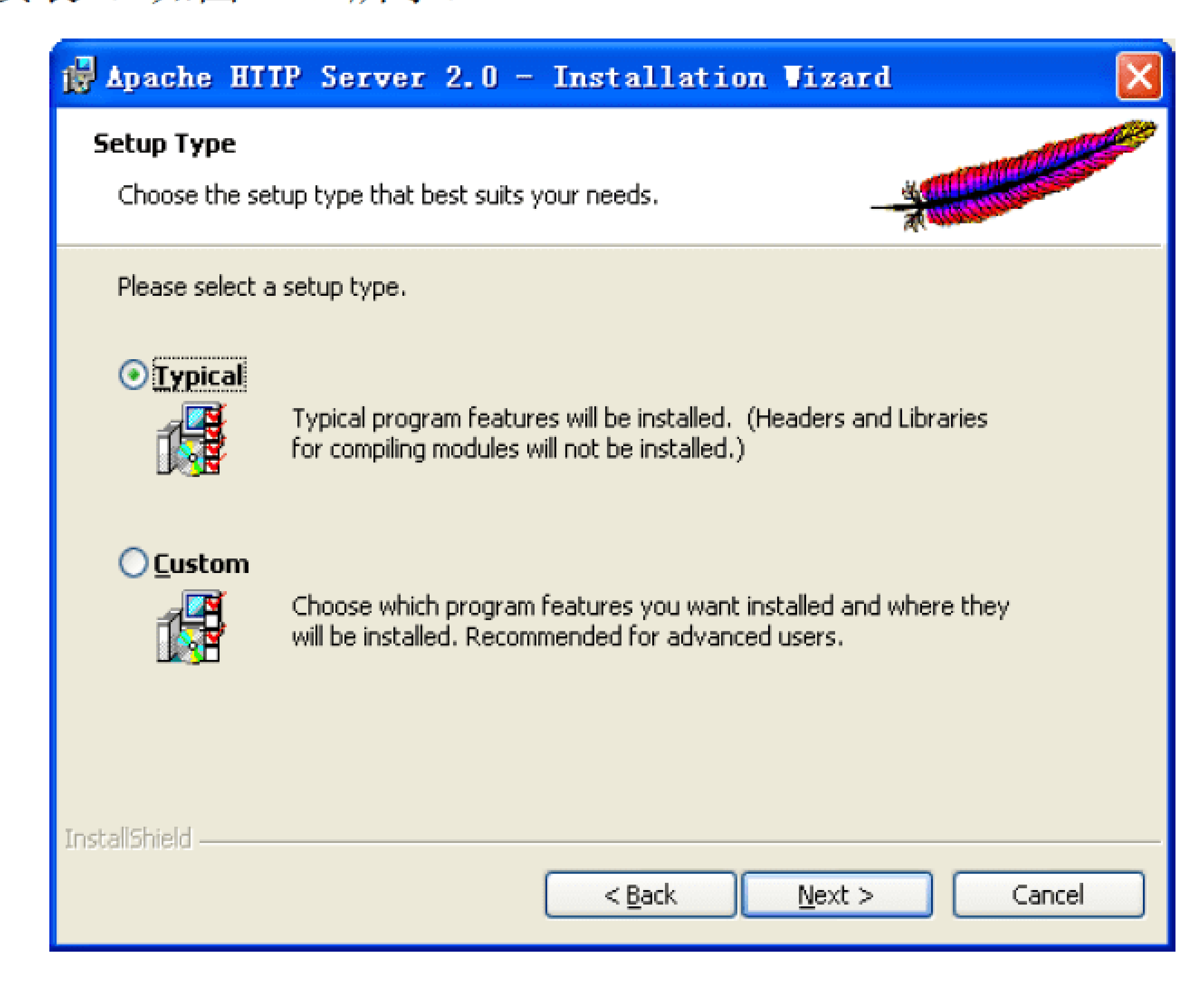


图 3-25　选择安装类型

继续单击 Next 按钮，出现 Apache 安装位置选择窗口。Apache 默认被安装到 C:\Program Files\Apache Group\目录下。如果你希望安装在其他位置，可以单击 Change 按钮来选择另外一个位置。这里采用默认位置，单击 Next 按钮，这时出现“安装准备已就绪”窗口。如果不需要对前面进行过的步骤作修改，就可以直接单击 Install 按钮开始安装 Apache。安装开始后会出现安装进度条，如图 3-26 所示。

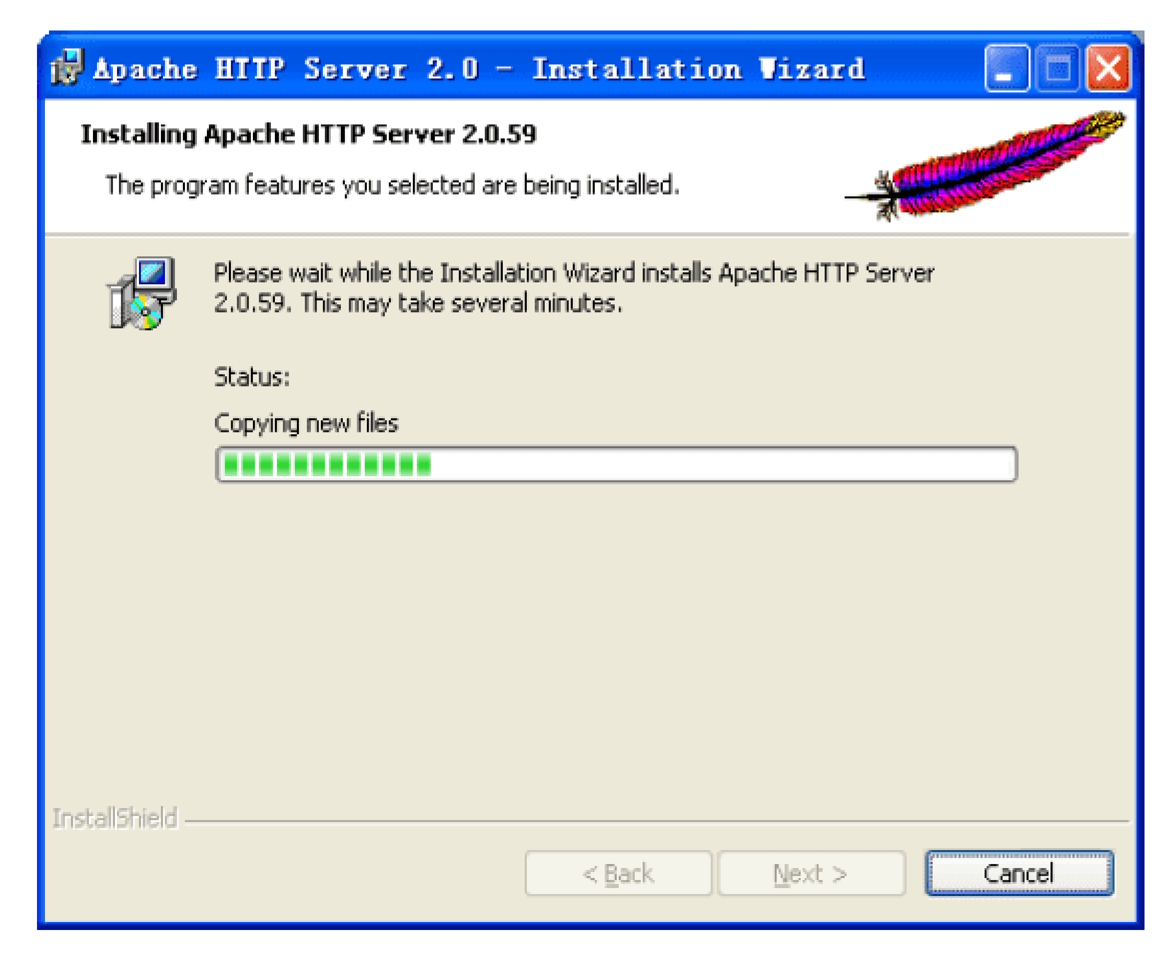


图 3-26　Apache 安装进度

全部安装完成后，会出现安装成功的提示窗口，单击 Finish 按钮结束安装程序，Apache 的安装完成了！为了验证安装是否成功，打开浏览器，在地址栏中输入 http://localhost/或者 http://127.0.0.1/，这时如果看到如图 3-27 所示的页面，那么说明 Apache 已经成功安装并开始服务了！



图 3-27　Apache 安装测试页面

安装成功后在系统程序组中会出现一个名为 Apache Http Server 2.0.59 程序组，其中包含了对 Apache 服务器的一些控制功能。如配置 Apache 参数、启动、关闭、重新启动 Apache

服务以及帮助信息等。

最后还要强调一点：虽然绝大多数情况下都可以快速顺利地安装成功 Apache，但也不排除安装失败的情况。由于操作系统版本、机器软件环境等影响，有可能在安装 Apache 的最后阶段会出现错误，或者安装之后无法启动。这时应根据 Apache 给出的错误提示查找并解决出错的原因。常见的错误有找不到 Apache 2 服务、端口冲突等。

对于找不到 Apache 2 服务，说明 Apache 没有成功地被安装为 Windows 服务，这时候可以手工启动 Apache，也可以在命令行模式下将其注册为服务。端口冲突一般是由于安装了其他软件占用了 80 端口所致，可以通过卸载无关软件或者修改 Apache 服务端口的办法解决。如果机器安装了防火墙等软件，也有可能导致 Apache 无法打开端口而无法启动。总之，如果遇到安装问题，可以通过阅读 Apache 的安装说明文档或者上网检索寻求解决办法。

3.4.4　将 PHP 与 Apache 建立关联

虽然 Apache 目前已经可以正常运行，并能提供静态网页服务，但此时它仍无法运行 PHP 网页。同前面讲过的 IIS 一样，要想让 Apache 能够运行 PHP 网页，也必须将 PHP 与 Apache 建立关联。

在将 PHP 与 Apache 建立关联之前，首先应该保证 PHP 已经被成功地安装。安装过程参考本章 3.3.2 节。接下来的所有工作都是建立在此基础之上的。PHP 安装好之后就可以按照以下步骤对 Apache 进行配置。

1. 找到 Apache 配置文件

同 IIS 的图形界面不同的是，Apache 的设置是通过修改其配置文件来实现的。这个配置文件的名字为 httpd.conf，这是一个纯文本文件，可以直接用记事本程序打开并编辑它。这个文件被存放在 Apache 安装目录的 Apache2\conf\目录下。也可以通过 Apache 程序组中的 Configure Apache Server 子程序组下面的 Edit the Apache httd.conf configuration file 菜单来直接打开这个配置文件。如图 3-28 所示。

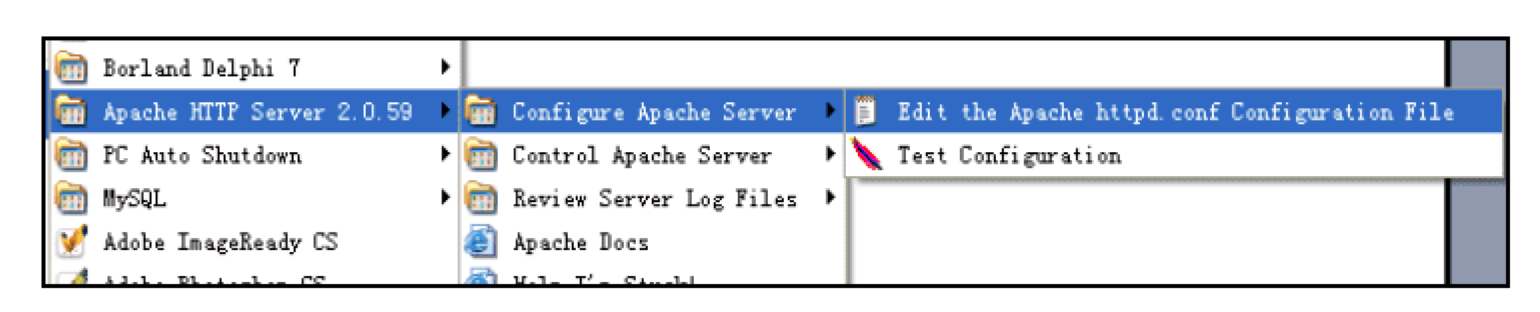


图 3-28　打开 APACHE 配置文件

2. 修改 Apache 主目录

打开 httpd.conf 之后，首先要做的就是设置网站的主目录。也就是默认情况下网页存放的位置。默认值为 Apache 安装目录的 Apache2\htdocs\下。如果把 Apache 安装在了 C:\Program files\Apache Group\，那么默认的主目录就是 C:\Program Files\Apache Group\Apache2\htdocs\。可以修改这个目录来适应自己的需要。比如在 D 盘创建一个目录 wwwroot，可以通过以下方法将此目录设置为主目录。

在 httpd.conf 中找到 DocumentRoot 参数，将其值修改为 D:/wwwroot/。如图 3-29 所示。从图中可以看出，Apache 的配置文件与 PHP 的在形式上有所不同。PHP 配置文件中以“=”

作为参数名和参数值的分隔符，而 Apache 中使用空格。还有一点就是 PHP 配置文件中表示路径时目录分隔符用“\”，而 Apache 中用“/”，这一点必须注意，不要因为粗心大意导致配置失败。



图 3-29 修改 APACHE 的主目录

3. 向 Apache 中加入 PHP 模块

有两种方式可以将 PHP 与 Aapche 建立关联，一种是 CGI 方式，另一种是模块方式。这里采用的是模块方式。在 httpd.conf 中任意位置插入以下两句代码：

```
LoadModule  php5_module  "c:/php5/php5apache2.dll"
AddType  application/x-httpd-php  .php
```

第一句代码的作用是使得 Apache 在启动时载入 PHP 模块。第二句的作用是使 Apache 能够识别.php 的扩展名。必须注意的是第一条语句中的 php5apache2.dll，容易错写成 php5apache.dll，因为在 PHP 的安装目录下这两个文件都存在。之所以这里要使用 php5apache2.dll 是因为我们采用的 Apache 版本为 2.0.59，属于 Apache 2.0.x 系列。如果使用的 Apache 版本是 1.3.x，那么此处就应该使用 php5apache.dll。

4. 指定 php.ini 文件的存放路径

Apache 中还有另外一个重要参数，用来指定 php.ini 文件的存放位置。由于 PHP 安装在了 c:\php5，php.ini 的位置就是 c:\php5\php.ini。在 httpd.conf 中任意位置加入下面一条语句：

```
PHPIniDir "C:/php5"
```

或者

```
PHPIniDir "C:/php5/php.ini"
```

这样一来，PHP 便知道到何处去搜索 php.ini，这样 php.ini 中的设置才能生效。添加完成后如图 3-30 所示。



图 3-30 修改 httpd.conf

上面四个步骤进行完之后，保存 httpd.conf，然后重新启动 Apache 使设置生效。重新启动的办法是在 Apache 程序组的 Control Apache Server 中选择 Restart，稍等之后 Apache 便重新启动完成。PHP 和 Apache 的关联到此为止便大功告成了！

下面我们完全可以编写一个例子，来测试 Apache 服务器是否可以正常运行 PHP 程序。不妨还是使用本章 3.3.5 小节中的测试代码，将其保存到 D:\wwwroot 下，然后在浏览器中输入 http://localhost/3-1.php，这时应当可以看到与图 3-16 几乎完全相同的输出结果。只不过表中的个别参数与使用 IIS 时略有不同。这表明 Apache 2+PHP 5 服务器环境搭建成功！

3.5　案例 3 将本地机器配置成 PHP 服务器并显示 PHP 配置信息

3.5.1　案例分析

本案例要求读者将自己使用的本地计算机配置成为 PHP 服务器，并编写程序显示 PHP 配置信息。对于此案例，应首先根据自己本地计算机所使用的操作系统来选择 PHP 版本和服务器软件。如果使用 Windows 操作系统，则一般选用 IIS 或 Apache 作为服务器软件。然后按照所学内容，安装配置 PHP，然后将 PHP 与服务器软件建立关联。最后书写一个简单 PHP 程序来输出 PHP 配置信息。

3.5.2　实施步骤

参照本章所讲内容，完成以下步骤：

（1）下载合适的 PHP 版本；
（2）按步骤完成 PHP 的安装工作；
（3）下载安装服务器软件；
（4）将服务器软件与 PHP 建立关联；
（5）设置好本地服务器的主目录和虚拟目录；
（6）编写一个 PHP 程序并运行，输出 PHP 配置信息。

3.6　本 章 小 结

本章中着重介绍了 PHP 服务器环境搭建的方法。并以 PHP+IIS 和 PHP+Apache 为例详细讲解了 Windows 平台上的 PHP 服务器搭建过程。对 PHP 的配置和 Apache 的配置也作了一定介绍。尤其是主目录和虚拟目录的设置。读者通过本章的学习，应能熟练掌握 PHP 与 IIS/Apache 搭配的服务器配置方法。

3.7 练 习 题

1. 试描述 PHP 程序的运行流程。
2. 目前常用的服务器软件有哪些？
3. 如何将 PHP 与 Apache 建立关联？
4. 主目录下面的子目录和虚拟目录有何不同？

CHAPTER 4

PHP 5 的基本语法

本章重点

- PHP 5 的程序结构
- PHP 5 的变量和常量
- PHP 5 中的运算符和表达式
- PHP 5 中的流程控制
- PHP 5 的自定义函数
- PHP 5 的类与对象

PHP + MySQL

4.1 PHP 语法概述

4.1.1 一个简单的 PHP 程序

在第 3 章中，已经搭建起了 PHP 的运行平台，下面就来学习 PHP 的基本语法。hello,world! 几乎已经变成了所有程序语言的第一个范例，在这儿也不例外，我们就先用 PHP 来写一个输出 hello,world!的简单的 PHP 程序。

程序清单 文档 4-1.html

```
1: <!—文件 4-1.php：一个 php 的简单程序-->
2: <html>
3:  <head>
4:  <title>First program</title>
5:  </head>
6:      <body>
7:      <?php
8:          echo "hello,world!";
9:      ?>
10:     </body>
11: </html>
```

这个 11 行的程序在 PHP 中不需经过编译等复杂的过程，只要将它放在配置好 PHP 平台的服务器中，并将它存成 4-1.php。在用户的浏览器端，只要在地址栏中输入 http://localhost/phpsource/chapt04/4-1.php（phpsource 为存放 php 文件的文件夹的服务虚拟目录，chapt04 为该目录下的一个文件夹，4-1.php 为 chapt04 文件夹下的一个 PHP 文件），就可以在浏览器上看到如下的效果（见图 4-1）。

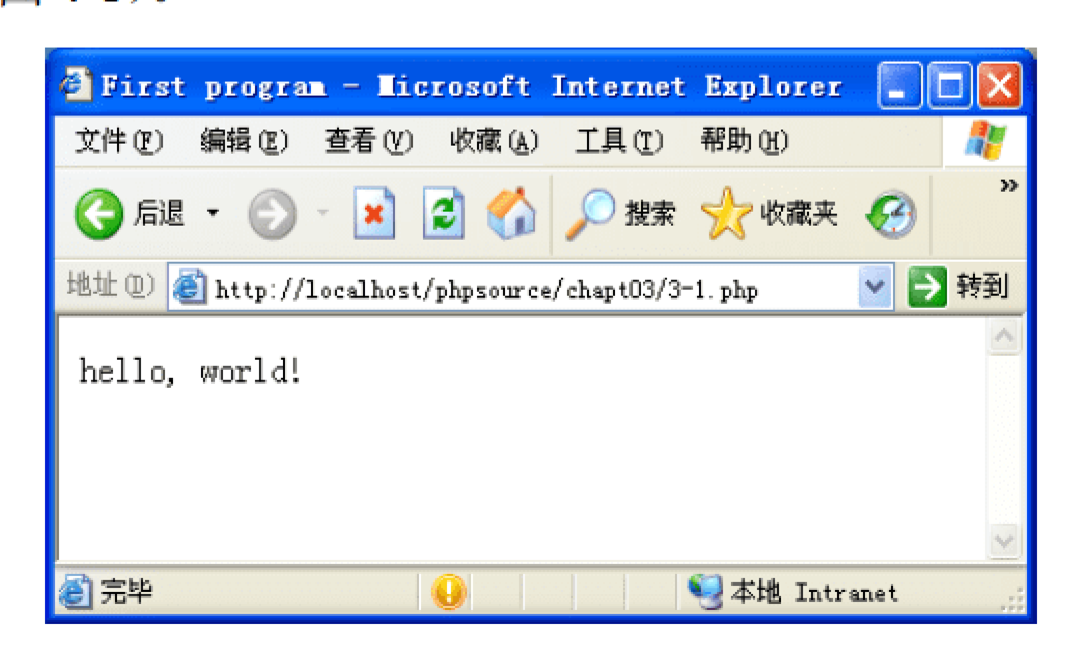


图 4-1 4-1.php 程序的运行结果

可以看到，这个程序只有 3 行有用，其他 6 行都是标准的 HTML 语句。第 7 行和第 9 行，分别是 PHP 的开始及结束的嵌入符号。第 8 行才是服务器端执行的程序。我们再通过浏览器窗口的“查看”→“源文件”操作来查看其源文件如下：

```
<!--文件 4-1.php：一个 php 的简单程序-->
<html>
     <head>
         <title>First program for PHP</title>
     </head>
     <body>
        hello,world!
    </body>
</html>
```

可以看出 PHP 程序在返回浏览器时和 JavaScript 完全不一样，PHP 的源程序没有传到浏览器，只在浏览器上看到短短的几个字 hello,world!。

4.1.2　PHP 嵌入 HTML 文档的方式

在文件 4-1.php 中“<?php”和“?>”为 PHP 的分界符，表示其中所包含的代码是 PHP 代码，当服务器读到该段程序的时候就会调用 PHP 的编译程序。将 PHP 代码嵌入到 HTML 文档中共有 4 种方式，具体如下：

1. 利用分界符“<?php”和“?>”

这是 PHP 最为普通的嵌入方式，也是 PHP 标准的嵌入方式，举例如下：

```
<?php
echo("这是一个标准方式的 PHP 语言的嵌入范例");
?>
```

强烈建议使用此方式，这种写法可以使程序在跨平台使用时减少不必要的麻烦！

2. 利用分界符“<?”和“?>”

这种方式是简写方式，必须在 php.ini 文件中将 shor_open_tag 设置为 On（PHP 5 中默认设置为 On），否则编译器将不予解析。使用举例如下：

```
<?
echo("这是一个简写方式的 PHP 语言嵌入范例");
?>
```

3. 利用分界符“<script language=“php”>”和“</script>”

这是类似于 JavaScript 风格的嵌入方式，对熟悉 NetScape 服务器产品的 Webmaster 人员而言，有相当的亲切感，举例如下：

```
<script language="php">
echo("这是类似 JavaScript 和 VBScript 风格的 PHP 语言嵌入范例");
</script>
```

4. 利用分界符“<%”和“%>”

这是一种具有 ASP 风格的嵌入方式，必须在 php.ini 文件中设置 asp_tag 为 On，否则编译器将不予解析。举例如下：

```
<%
echo("这是类似 ASP 风格的 PHP 语言嵌入范例");
%>
```

强烈建议少用这种方法，因为当 PHP 与 ASP 源代码混在一起时就麻烦喽！

提 示

> 其实，编写 PHP 程序最好的方法，是先处理好纯 HTML 格式的文件之后，再将需要变量或其他处理的地方改成 PHP 程序。这种方法，可以让您在开发上达到事半功倍的效果。

下面的程序 4-2.php 就使用了 4 种不同的 PHP 嵌入方式：

程序清单　文档 4-2.php

```
<!--文件 4-2.php：PHP 四种不同的嵌入方式-->
<HTML>
 <HEAD>
     <TITLE>PHP 不同风格的分界符</TITLE>
 </HEAD>
 <BODY>
     <?php
         echo "用\"&lt?php\" 和 \"?&gt\" 做分界符";
     ?>
     <p>
     <?
         echo "用\"&lt?\" 和 \"?&gt\" 做分界符";
     ?>
     <p>
     <script language="php">
             echo "用\"&ltscript language=\"php\"&gt\" 和
         \"&lt/script&gt\" 做分界符";
     </script>
     <p>
     <%
         echo "用\"&lt%\" 和 \"%&gt\" 做分界符";
     %>
 </BODY>
</HTML>
```

其运行结果如图 4-2 所示。

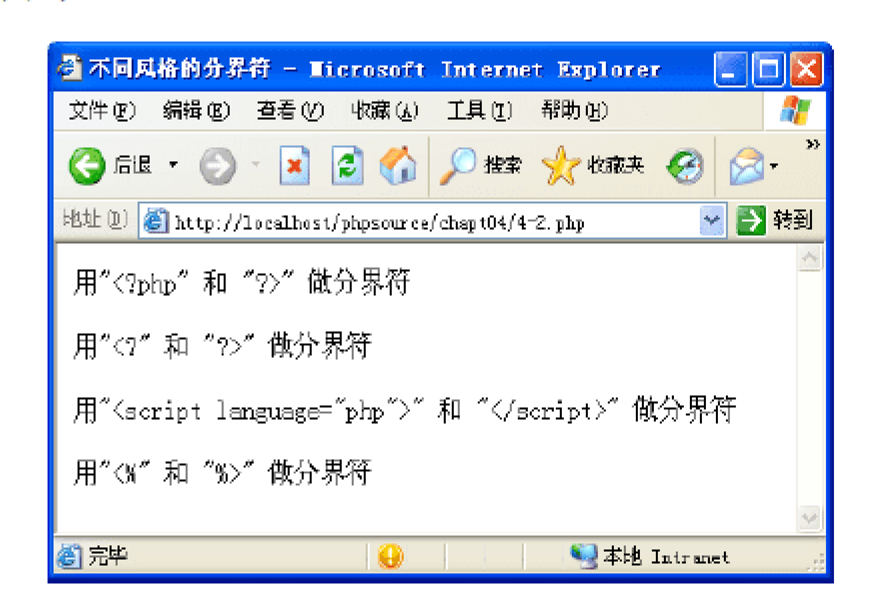


图 4-2　4-2.php 程序的运行结果

提 示

> 要使上面的程序能运行输出如图 4-2 所示的结果，一定要在 php.ini 中设置 short_open_tag=On 和 asp_tag=On，否则将得不到如图 4-2 所示的结果。

4.1.3　PHP 程序的注释方法

在 PHP 的程序中，加入注释的方法很灵活。可以使用 C 语言、C++ 语言或者是 Unix 的 Shell 语言的注释方式，也可以混合使用。具体方法如下：

- "//"：这是从 C＋＋语法中借鉴来的，该符号只能注释一行。
- "/*" 和 "*/"：这是 C 语言的注释符，符号之间的都将被注释掉。
- "#"：这是 UNIX 的 Shell 语言风格的注释符，也只能注释一行。

下面的程序 4-3.php 中就用了 3 种不同风格的注释：

程序清单　文档 4-3.php

```
<!--文件 4-3.php:php 的三种不同的注释方法-->
<HTML>
  <HEAD>
      <TITLE>PHP 不同风格的注释</TITLE>
  </HEAD>
  <BODY>
      <?php
          echo "使用//注释单行<p>"; //单行注释,看我能输出否
      echo "使用/**/注释一段<p>";   /*段注释,
                                     我还是不能输出，唉......
                                     可注释一段*/
          echo "使用 Unix Shell 风格的#注释";    #"#"也只能注释一行
      ?>
   </BODY>
</HTML>
```

其运行结果如图 4-3 所示。

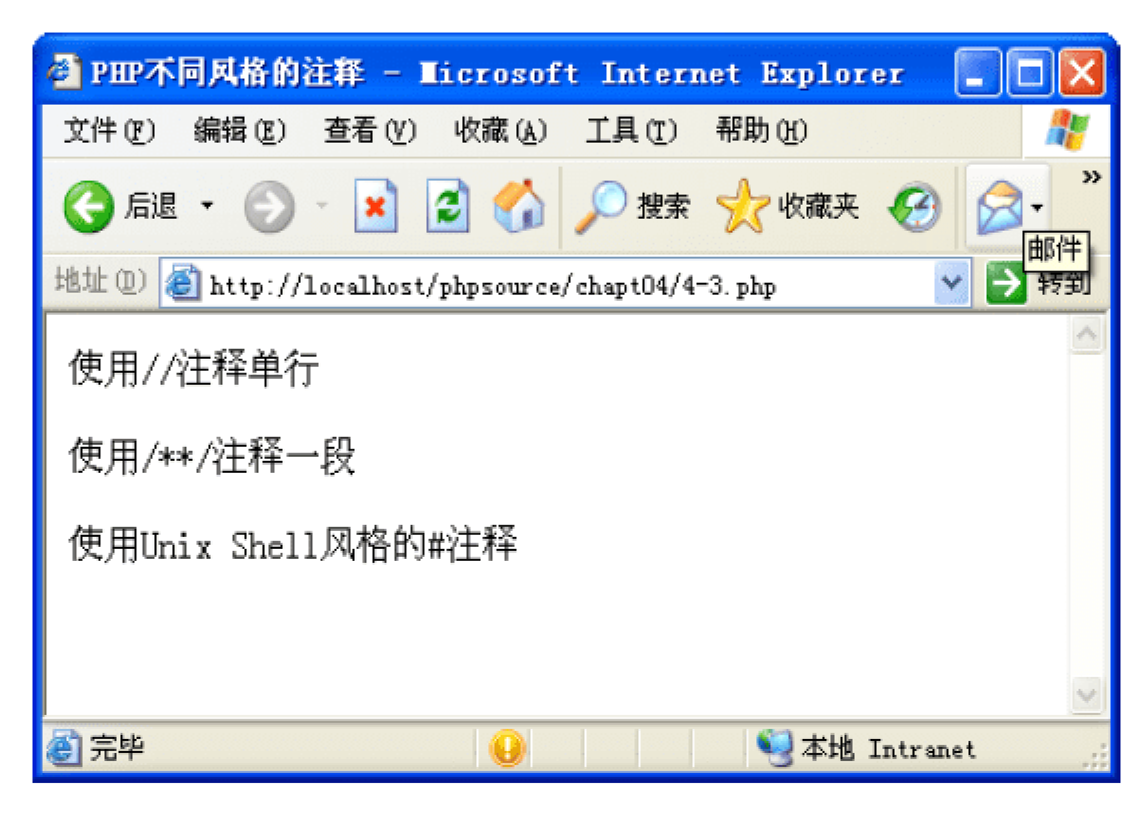


图 4-3　4-3.php 程序的运行结果

4.1.4　PHP 的文件引用

PHP 最吸引人的特色之一大概就是它的文件引用了。用这个方法可以将常用的功能写成一个函数，放在文件之中，然后引用之后就可以调用这个函数了。

引用文件的方法有两种：require 和 include。两种方法提供不同的使用弹性。

（1）require 的使用方法 require 的调用方法如下：

```
require("MyRequireFile.php");
```

这个函数通常放在 PHP 程序的最前面，PHP 程序在执行前，就会先读入 require 所指定引入的文件，使它变成 PHP 程序网页的一部份。常用的函数，亦可以用这个方法将它引入网页中。

（2）include 使用方法 include 的调用方法如下：

```
include("MyIncludeFile.php");”
```

这个函数一般是放在流程控制的处理部分中。PHP 程序在读到 include 的文件时，才将它读进来。这种方法，可以使程序执行流程简单化。

下面先建立一个名为 4-4.php 的文件，输入如下代码：

程序清单　文档 4-4.php

```
<!--文件 4-4.php：php 文件的引用-->
<html>
    <head>
        <title>PHP 文件的引用</title>
    </head>
    <body>
        <?php
            echo "这是主文件"4-4.php”输出的！<br>";
```

```
        include("include.msp");
                //引用同目录下名为"include.inc"的 php 文件
        ?>
    </body>
</html>
```

然后我们再建立一个名为"include.msp"的文件，其代码如下：

```
<!--文件 include.msp：被"4-4.php"文件所引用的引用-->
<?php
    echo "这是从"include.msp"文件中输出的！";
?>
```

其运行结果如图 4-4 所示。

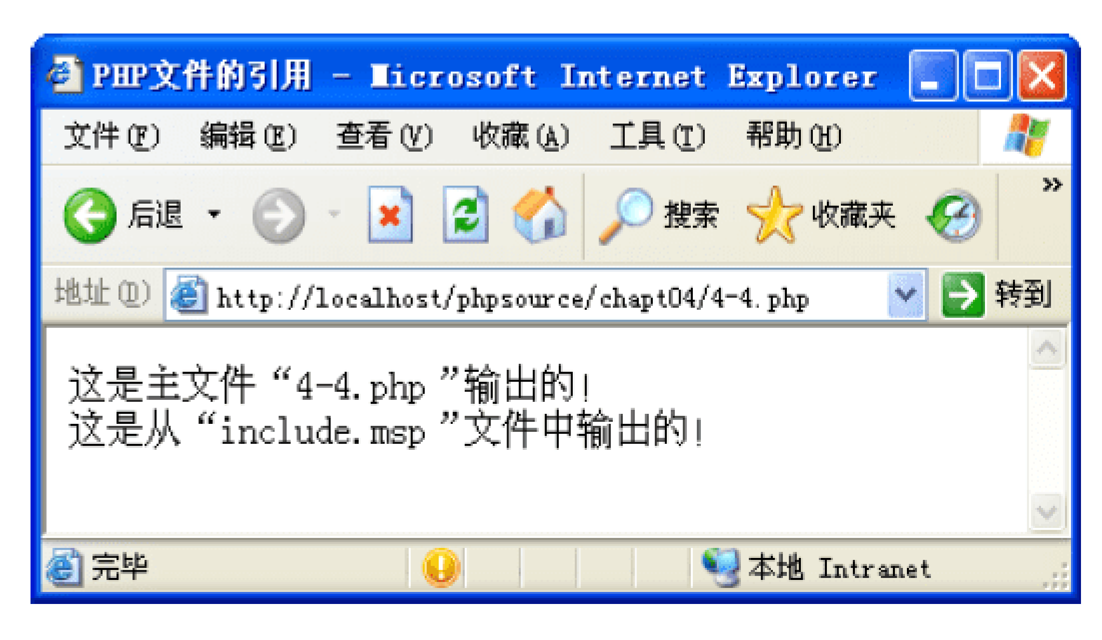


图 4-4　4-4.php 程序的运行结果

提 示

include.msp 为被引用的文件名，只要保证该文件的类型为文本类型，它的名字可以任意命名，包含其后缀。这样你是不是可以做出很有个性的文件呢？

4.2　PHP 的常量和变量

4.2.1　PHP 的常量类型

PHP 的常量包含两种：一种是系统预定义常量，另一种是自定义常量。

PHP 中预定了很多系统常量，这些系统常量可以很好地帮助我们了解系统当前的情况，而且这些常量可以随时调用。具体列举如下：

1. _FILE_

这个默认常量是 PHP 程序文件名。若引用文件（include 或 require），则在引用文件内的该常量为引用文件名，而不是引用它的文件名。

2. _LINE_

这个默认常量是 PHP 程序行数。若引用文件（include 或 require），则在引用文件内的

该常量为引用文件的行，而不是引用它的文件行。

3. PHP_VERSION

这个内建常量是 PHP 程序的版本，如 3.0.8-dev。

4. PHP_OS

这个内建常量指执行 PHP 解析器的操作系统名称，如 Linux。

5. TRUE

这个常量就是真值 （true）。

6. FALSE

这个常量就是伪值（false）。

7. E_ERROR

这个常量指到最近的错误处。

8. E_WARNING

这个常量指到最近的警告处。

9. E_PARSE

本常式为解析语法有潜在问题处。

10. E_NOTICE

这个常量经常发生不寻常但不一定是错误处。例如存取一个不存在的变量。这些 E_ 开头形式的常量，可以参考 error_reporting()函数，有更多的相关说明。

下面的例子就是利用系统预定义变量输出一些系统参数，具体内容如下：

程序清单　文档 4-5.php

```
<!--文件 4-5.php：PHP 预定义常量-->
<HTML>
 <HEAD>
     <TITLE>PHP 预定义常量</TITLE>
 </HEAD>
 <BODY>
     <?php
         echo(_LINE_);    //输出 8
         echo "<p>";
         echo(_FILE_);
         echo "<p>";
         echo(_LINE_);    //输出 12
         echo "<p>";
         echo PHP_VERSION;
```

```
            echo "<p>";
            echo PHP_OS;
        ?>
</BODY>
</HTML>
```

其运行结果如图 4-5 所示。

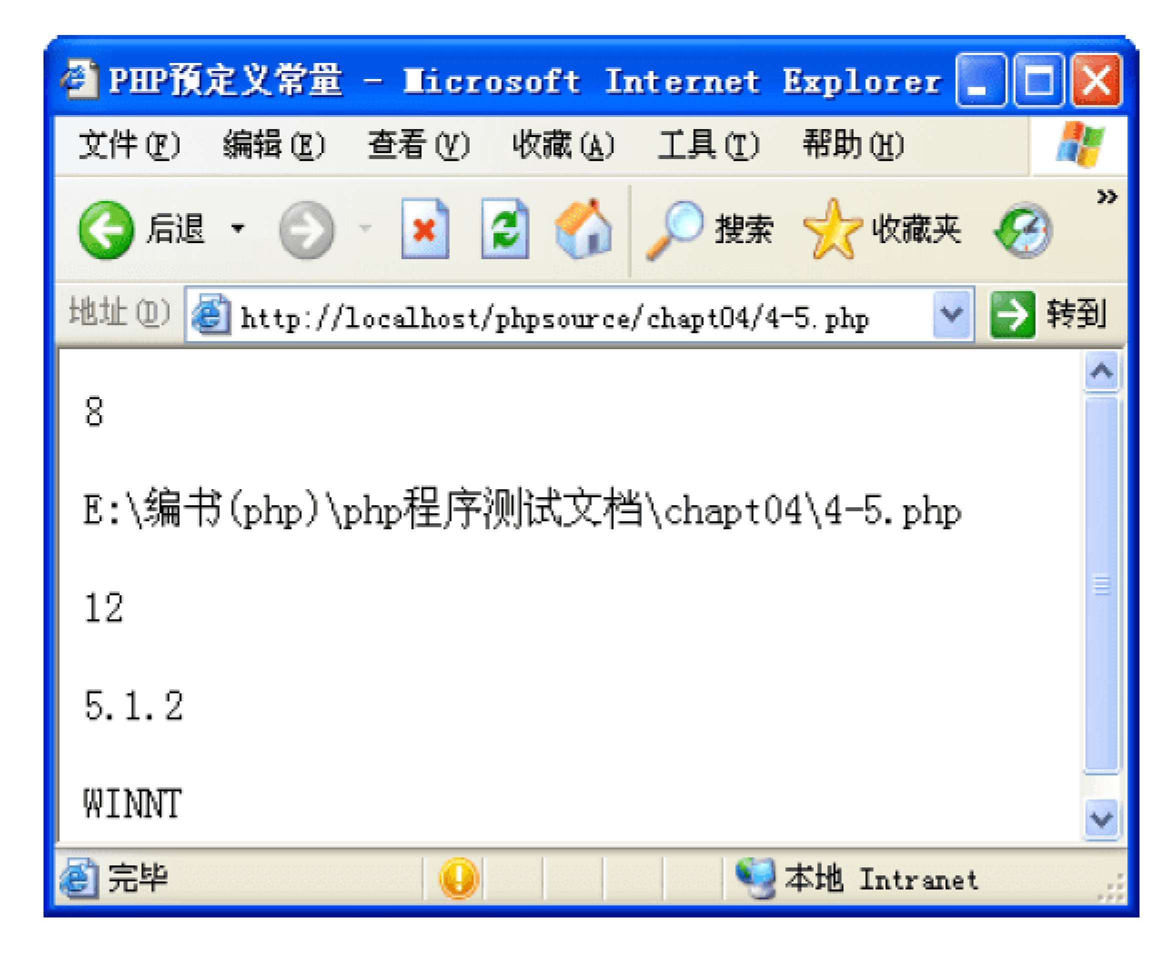


图 4-5　4-5.php 程序的运行结果

当然在实际编写程序时，仅使用以上的系统预定义常量是不够用。define()可以自行定义所需要的常量。用法详见下例：

程序清单　文档 4-6.php

```
<!--文件 4-6.php: PHP 自定义常量-->
<HTML>
 <HEAD>
     <TITLE>PHP 定义常量</TITLE>
 </HEAD>
 <BODY>
     <?php
         define("COPYRIGHT","Copyright &copy; 2006,www.rzpt.cn");
         echo COPYRIGHT;
         ?>
</BODY>
</HTML>
```

其运行结果如图 4-6 所示。

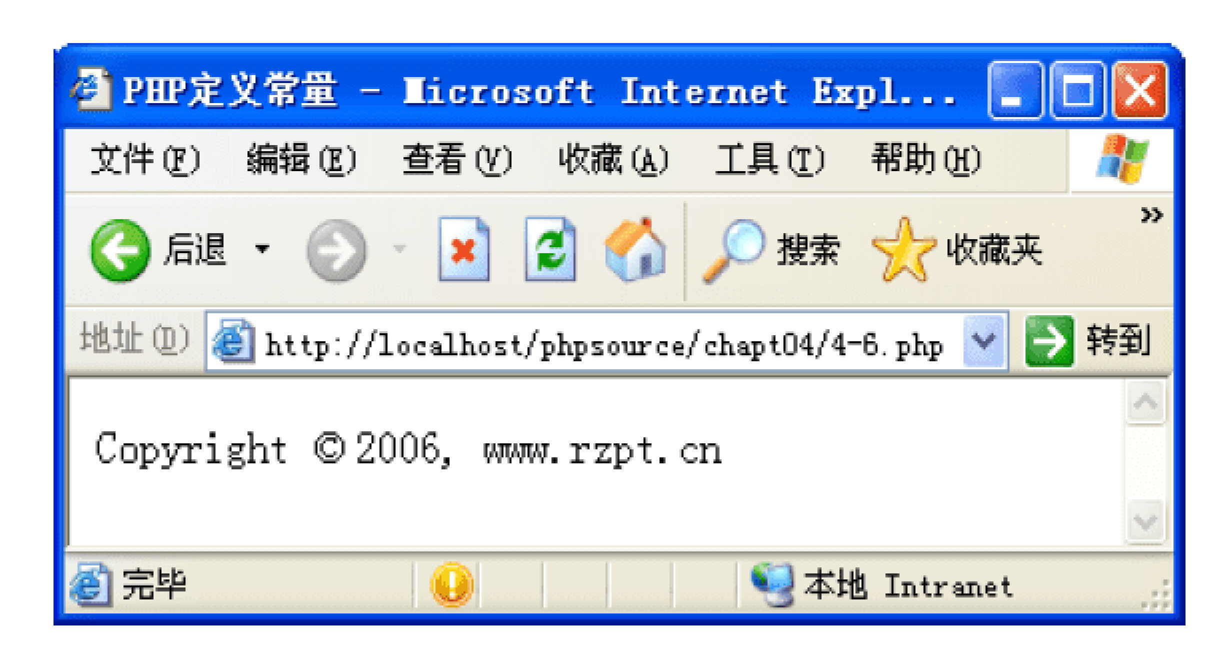


图 4-6 4-6.php 程序的运行结果

4.2.2 PHP 的变量类型

PHP 中预先定义了很多变量，可以随时在脚本中引用。PHP 的预定义变量将在后面的章节和附录中涉及到。下面主要讲解自定义变量。

PHP 的变量类型不多，有以下 6 种：

- string：字符串型。
- integer：整型。
- double （floating point number）：双精度型（浮点型）。
- boolean：布尔型。
- array：数组。
- object：对象。

PHP 的变量属于松散的数据类型，具体使用时应注意以下几点：

- 变量名要以“$”开头，且区分大小写；
- 变量不必预先定义或声明；
- 变量在使用时编译器可动态进行类型指定和转换；
- 变量如果未赋值而直接使用，变量值将被视为空。

提 示

> 最好使用相同的变量命名风格，以免在团队合作或自己查找错误的时候因变量大小的问题，浪费许多无谓的时间，那就得不偿失了。变量之间若要转换类型，当然是很自由的，但浮点数转成整数就有点牵强了，不过可以将浮点数转成字符串，也是很好处理的。

下面，就分别对这 6 种变量类型进行学习。

1. 字符串（string）

无论是单一字符或数千字的字符串都是使用这个变量类型。值得注意的是要指定字符串给字符串变量时，要在头尾都加上双引号或单引号（例如：“这是字符串” 或 "这是字符串"）。在 PHP 中提供了一些转移字符，用以辅助那些已经被程序语法结构占用了的特殊字

符，见表 4-1。

表 4-1　PHP 5 中的转义字符及其含义

转移字符	含义
\"	双引号
\\	反斜线
\n	换行
\r	回车
\t	制表符（TAB）
\$	美元符号（$）
\x 两位数字	表示十六进位字符

提 示

以上的转义字符 PHP 中 “'” 和 “"” 是不一样的，而且也可能无效，在使用时一定要测试通过了之后再用哦！再比如若 “\n” 换行无效，可以使用 “
” 来换行。

2. 整型（integer）

在 32 位的操作系统中，它的有效范围是-2 147 483 648 ~+2 147 483 647。要使用 8 进位整数可以在前面加 0（零），要使用 16 进位整数可以在前面加 0x，示例如下：

① $int1=1234　　//十进制正整数
② $int1=-1234　　//负整数
③ $int1=01234　　//八进制整数
④ $int1=0x1234　　//十六进制整数

3. 双精度型（浮点型）（double(floating point number)）

在 32 位的操作系统中，它的有效范围是 1.7E-308 ~ 1.7E+308。如：

① $float1=666.66
② $float2=6.6666e2　　//表示 6.6666 乘以 10 的 2 次方，为指数形式的浮点数

提 示

浮点型变量显示所用的十进制数的个数由 php.ini 文件中的 precision 定义，预定值为 12，即浮点数的最长为 14 个数字。

4. 布尔型（boolean）

布尔型也就是逻辑型，其值非真即假，只用于条件表达式和逻辑表达式中，用以控制程序流程。这种类型是从 PHP 4 中才开始存在的。通常 1 即为 true，0 为 false。

要将一个值转换成 boolean，用 bool 或者 boolean 来强制转换。但是很多情况下不需要用强制转换，因为当运算符，函数或者流程控制需要一个 boolean 参数时，该值会被自动转换。

有了 PHP 变量的理论知识下面来通过实例来看一下具体使用：

程序清单　文档 4-7.php

```
<!--文件 4-7.php：PHP 变量的使用=>字符串、整型、浮点型、布尔型-->
<HTML>
 <HEAD>
     <TITLE>PHP 变量的使用之一</TITLE>
 </HEAD>
 <BODY>
      <?php
          $string1="输出字符串变量类型的内容！";
          echo $string1;          //输出字符串变量“$string”的内容
           echo "<br>";            //输出换行
           $string2="输出特殊字符：";
           echo $string2."\\";
           echo "\$";
           echo '\'';
           echo "\"";
           echo "\x52";
           echo "<br>";
           $int1=01234;    //八进制整数
           $int2=0x1234;
           echo "输出整型变量的值：";
           echo $int1;       //输出 668
           echo "\t";        //输出一个制表位
           echo $int2;       //输出 4660
           echo "<br>";
           $float1=6.6666e2;
           echo "输出浮点型变量的值：";
           echo $float1;      //输出 666.66
           echo "<br>";
           echo "输出布尔型变量的值：";
           echo(boolean)($int1);     //输出转换后的布尔变量-"1"
       ?>
 </BODY>
 </HTML>
```

上面的例子涉及到了变量类型中的字符串、整型、浮点型、布尔型变量的使用，其运行结果如图 4-7 所示。

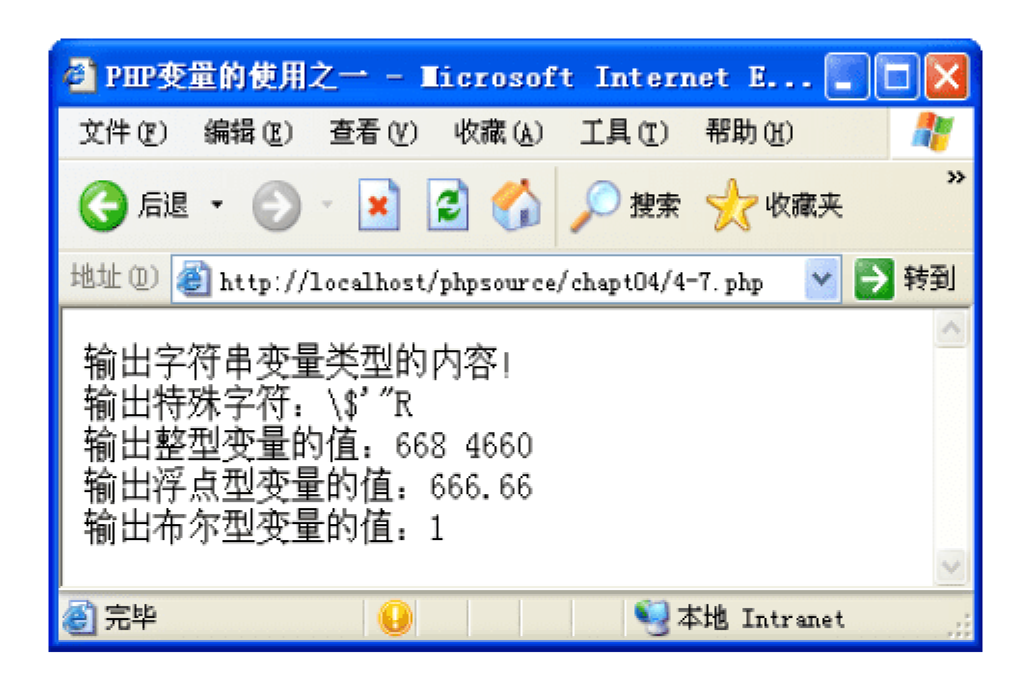


图 4-7 4-7.php 程序的运行结果

5. 数组（array）

数组变量可以是一维、二维或者更多维数组，其中的元素也很自由，可以是字符串、整型、浮点型、布尔型，甚至是数组或对象。PHP 还同时支持以数字为下标的索引数组（Scalar Array）和以字符串为下标的关联数组（Associative Array），这二者虽没有任何区别，但也算得上是 PHP 比别的语言特别的地方之一吧！在 PHP 中可以使用 list()函数或者 array()函数来创建数组，也可以直接进行赋值。

用 array()函数创建数组的方法如下例：

程序清单 文档 4-8.php

```
<!--文件 4-8.php：用 array()函数创建 PHP 数组-->
<HTML>
     <HEAD>
         <TITLE>用 array()函数创建 PHP 数组</TITLE>
     </HEAD>
     <BODY>
         <?php
         $arr=array
         (
             0=>6,
             2=>6.666e2,
             1=>"我爱 PHP",
                 "str"=>"string",
             );
             for($i=0;$i<count($arr);$i++)
             {
                 $print=each($arr);
             echo "$print[value]<br>";
         }
     ?>
</BODY>
</HTML>
```

程序中使用“=>”符号来给数组元素赋值，for 循环用来输出整个数组。其中函数 count()用来计算出数组元素的个数，函数 each()返回当前数组指针的索引/值对，在后面的章节还将会讲到。PHP 的下标只是作为以后存取的标识，没有任何符号意义，数组元素的排列顺序是以加入的先后顺序为准，故程序 4-8.php 的显示效果如图 4-8 所示。

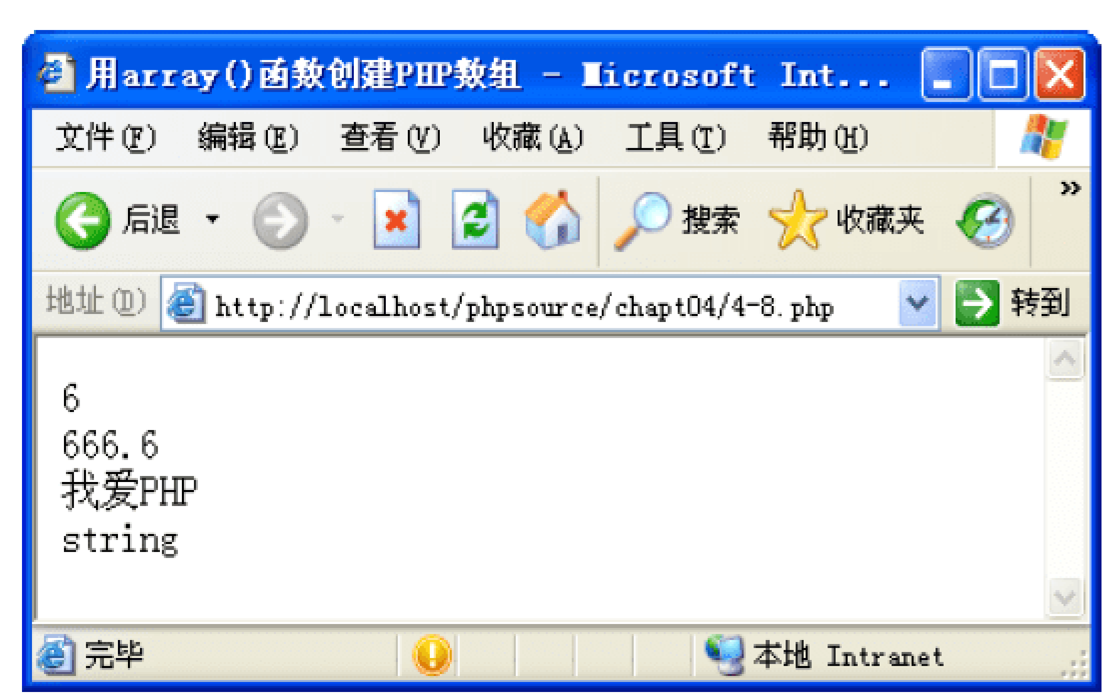


图 4-8　4-8.php 程序的运行结果

也可以采用将数组元素一个一个赋值的方法，例如：

程序清单　文档 4-9.php

```
<!--文件 4-9.php：逐一给数组元素赋值-->
<HTML>
 <HEAD>
     <TITLE>逐一给数组元素赋值</TITLE>
 </HEAD>
 <BODY>
     <?php
           $arr[0]=6;
           $arr[2]=6.666e2;
           $arr[1]="我爱 PHP";
           $arr["str"]="string";
         for($i=0;$i<count($arr);$i++)
         {
             $print=each($arr);
             echo "$print[value]<br>";
         }
     ?>
 </BODY>
</HTML>
```

其运行结果如图 4-9 所示。

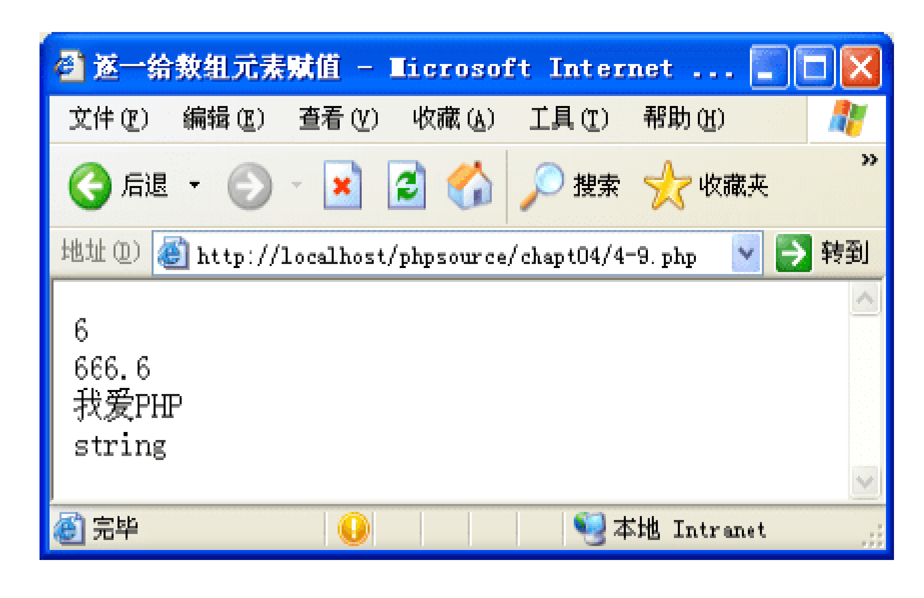


图 4-9　4-9.php 程序的运行结果

当然，还可以采用下面的更为简洁的方法赋值：

程序清单　文档 4-10.php

```
<!--文件 4-10.php：数组元素简洁赋值-->
<HTML>
 <HEAD>
     <TITLE>数组元素简洁赋值</TITLE>
 </HEAD>
 <BODY>
     <?php
        $arr=array(6,6.666e2,"我爱 PHP","string");
         for($i=0;$i<4;$i++)
             {
             echo $arr[$i]."<br>";
             }
     ?>
  </BODY>
</HTML>
```

当用上面这种简洁方式给数组赋值时，数组的默认下表为 0、1、2、3。其运行结果一样，如图 4-10 所示。

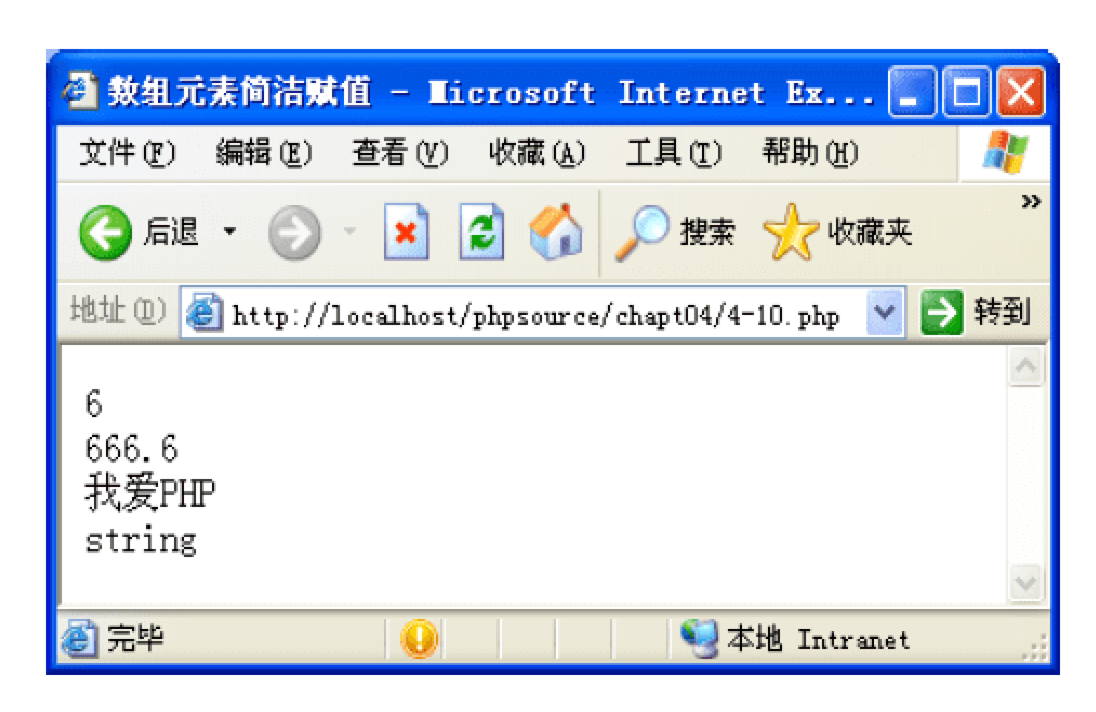


图 4-10　4-10.php 程序的运行结果

PHP 中多维数组与一维数组的区别在于多维数组有两个或多个下标，它们的用法基本上是一样的。下面的例子就是采用逐一给二维数组元素赋值的方法来创建和使用二维数组。

程序清单　文档 4-11.php

```
<!--文件 4-11.php：多维数组的逐一赋值法-->
<HTML>
     <HEAD>
         <TITLE>多维数组的逐一赋值</TITLE>
     </HEAD>
     <BODY>
         <?php
         $arr[0][0]=6;
         $arr[0][1]=6.666e2;
         $arr[1][0]= "我爱 PHP";
         $arr[1]["str"]="string";
         for($i =0;$i<count($arr);$i++)
         {
             for($j=0;$j<count($arr[$i]);$j++)
             {
                 $print=each($arr[$i]);
                 echo "$print[value]<br>";
             }
         }
     ?>
</BODY>
</HTML>
```

程序执行过程不同，但运行结果还是一样，如图 4-11 所示。

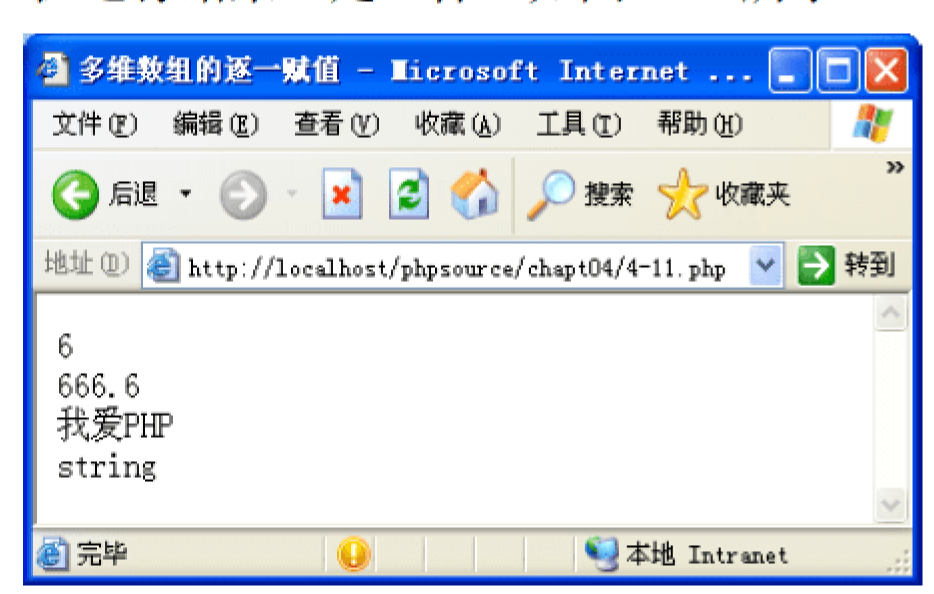


图 4-11　4-11.php 程序的运行结果

我们还以使用层次更明显、更容易理解和接受的多维数组赋值方式：嵌套的 array()函数方式来创建 PHP 数组并给数组元素赋值。具体使用如下例：

程序清单　文档 4-12.php

```
<!--文件 4-12.php：用嵌套的 array()函数创建 PHP 数组-->
<HTML>
 <HEAD>
```

```
    <TITLE>用嵌套 array()函数创建 PHP 数组</TITLE>
</HEAD>
<BODY>
    <?php
      $arr=array
     (
           0=>array
          (
             0=>6,
             2=>6.666e2,
          ),             //此处应该是","，而不是";"
           1=>array
          (
             0=>"我爱 PHP",
            "str"=>"string"
          )
          );
          for($i =0;$i<count($arr);$i++)
          {
             for($j=0;$j<count($arr[$i]);$j++)
             {
                $print=each($arr[$i]);
                echo "$print[value]<br>";
             }
          }
        ?>
    </BODY>
</HTML>
```

其运行结果如图 4-12 所示。

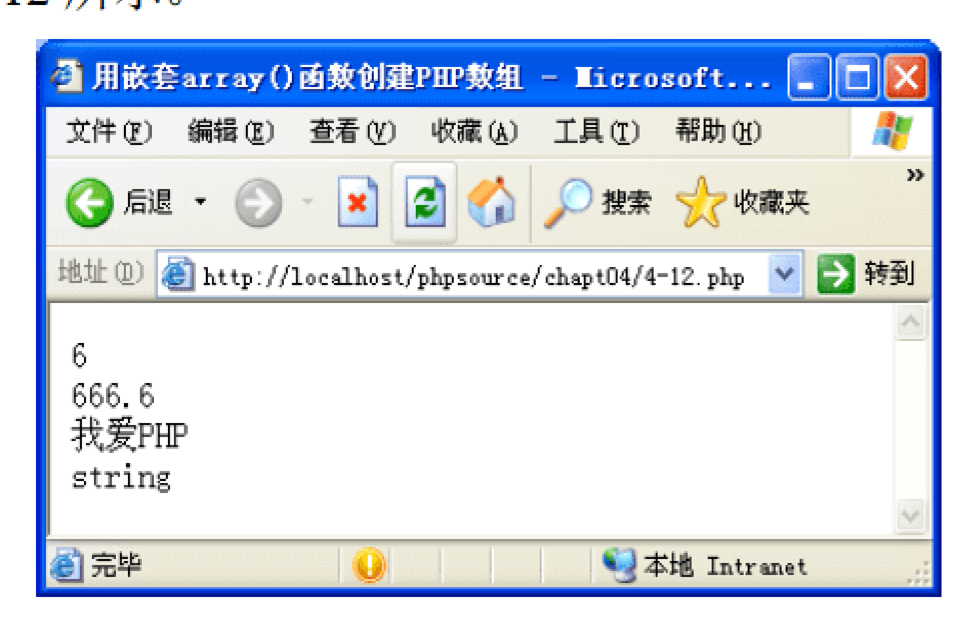


图 4-12　4-12.php 程序的运行结果

6. 对象（object）

object 为对象类型变量，目前在 PHP 中的默认类并不是太多，不过相信有了大家的努力一定会丰富起来的。我们将在 4.6 节中单独讲解对象的使用。

4.3 PHP 的运算符和表达式

4.3.1 PHP 的运算符

PHP 的运算符大部分是从 C 语言中借用而来的，共分以下几类：

① 算术运算符：+、–、*、/、%、++、--

② 字符串运算符：.

③ 赋值运算符：=、+=、-=、*=、/=、%=、.=

④ 位运算符：& 、| 、^ 、<< 、>> 、~

⑤ 逻辑运算符：&&(And)、||(Or)、xor(Xor)、!(Not)

⑥ 比较运算符：< 、> 、<= 、>= 、== 、===、!=

⑦ 其他运算符：$、&、@、-> 、=>、? :

下面就详细讲解各类运算符。

1. 算术运算符号

就是用来处理四则运算的符号，这是最简单，也最常用的符号，尤其是数字的处理，几乎都会使用到算术运算符号，其具体意义如表 4-2 所示：

表 4-2 算数运算符及其意义

符号	意义
+	加法运算
-	减法运算
*	乘法运算
/	除法运算
%	取余数
++	累加
--	累减

下面来具体看一下算术运算符的使用，如下例：

程序清单 文档 4-13.php

```
<!--文件 4-13.php：算术运算符的应用-->
<HTML>
<HEAD>
      <TITLE>算术运算符的应用</TITLE>
 </HEAD>
 <BODY>
      <?php
        $a=10;
```

```
        $b=3;
        echo $a."+".$b."=";
        echo $a+$b."<br>";
        echo $a."-".$b."=";
        echo $a-$b."<br>";
        echo $a."*".$b."=";
        echo $a*$b."<br>";
        echo $a."/".$b."=";
        echo $a/$b."<br>";
        echo $a."%".$b."=";
        echo $a%$b."<br>";
        echo $a."++=";
        echo $a++."<br>";
        $a=10;
        echo "++".$a."=";
        echo ++$a."<br>";
        $a=10;
        echo $a."--=";
        echo $a--."<br>";
        $a=10;
        echo "--".$a."=";
        echo --$a."<br>";
        $c="b";
      echo "\"b\"++=";
      $c++;
      echo "\"".$c."\"<br>";
    ?>
  </BODY>
</HTML>
```

其运行结果如图 4-13 所示。

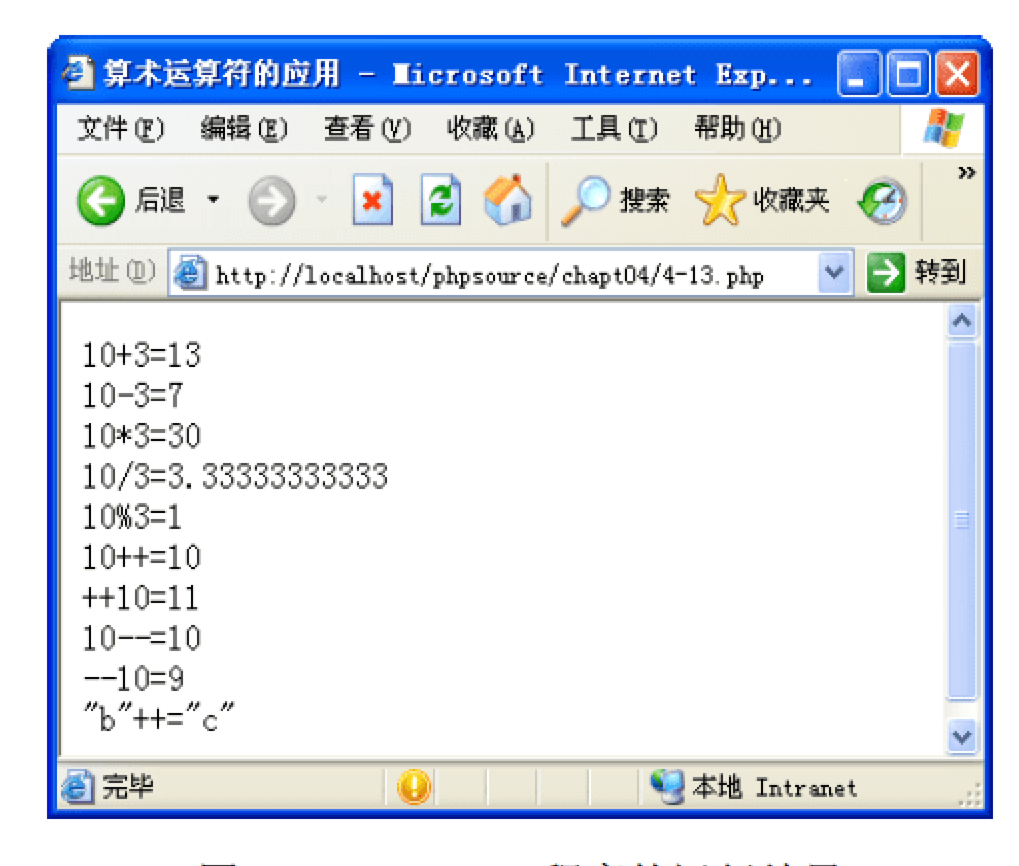


图 4-13　4-13.php 程序的运行结果

提 示

> PHP 中的除法运算得到的结果是整数就是整数，是实数就是实数。PHP 中的求模运算会对两操作数自动取整。另外，在 PHP 中字符也可以进行自增运算。

2. 字符串运算符

字符串运算符号只有一个，就是英文的句号“.”。它可以将字符串连接起来，变成合并的新字符串，也可以将字符串与数字连接，这时类型会自动转换，具体用法如下例：

程序清单　文档 4-14.php

```
<!--文件 4-14.php：字符串运算符的应用-->
<HTML>
 <HEAD>
     <TITLE>字符串运算符的应用</TITLE>
 </HEAD>
 <BODY>
     <?php
         $int1=2;
         $int2=0;
         $str1="PHP 5";
         $str2="功能强大";
         echo $str1.".".$int1.".".$int2.$str2." ==>我爱 PHP 5";
     ?>
 </BODY>
</HTML>
```

其运行结果如图 4-14 所示。

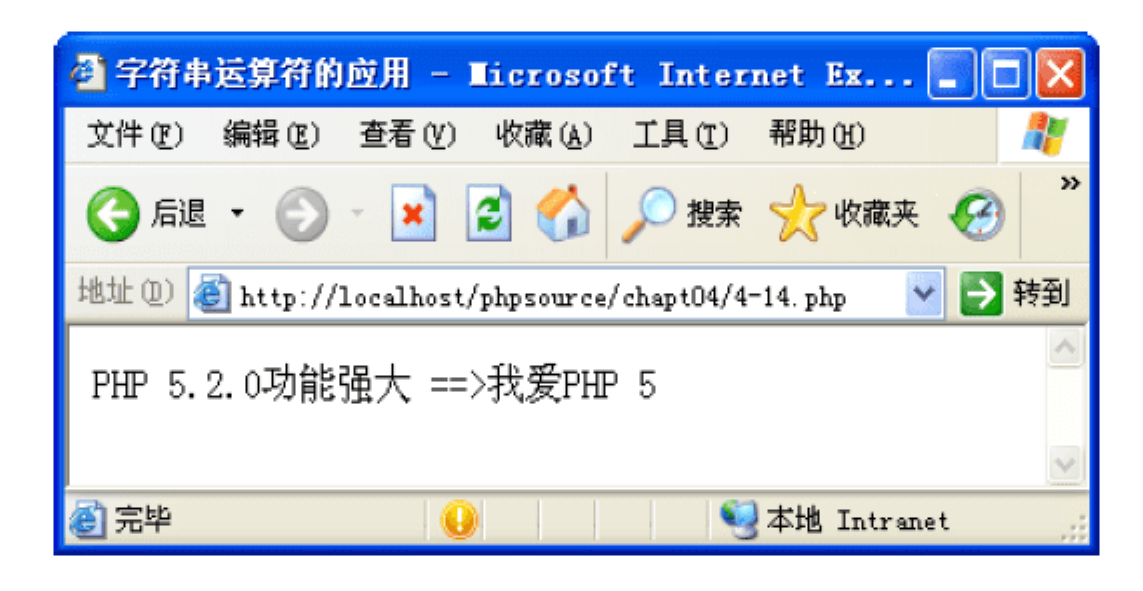


图 4-14　4-14.php 程序的运行结果

3. 赋值运算符

赋值运算符的具体意义如表 4-3 所示。

表 4-3　赋值运算符及其意义

符号	意义
=	将右边的值赋给左边的变量
+=	将左边的值加上右边的值赋给左边的变量
-=	将左边的值减去右边的值赋给左边的变量
*=	将左边的值乘以右边的值赋给左边的变量
/=	将左边的值除以右边值赋给左边的变量
%=	将左边的值对右边取余数赋给左边的变量
.=	将左边的字符串连接到右边

比如，“$a+=$b”等价于“$a=$a+$b”，其他赋值运算的等价关系可依此类推。赋值运算可以让程序更精简，增加程序的执行效率。

4. 位运算符

PHP 中的位运算符有 6 个，常用于二进制的运算场合，其具体含义如表 4-4 所示。

表 4-4　位运算符及其意义

符号	意义
&	按位与
|	按位或
^	按位异或
<<	按位左移
>>	按位右移
~	按位取反

其中"~"是单目运算符，其他的都是双目运算符。与、或、异或和取反的运算规则如下：

0&0=0　0&1=0　1&0=0　1&1=1　（与：有假就假，都真才真）
0|0=0　0|1=1　1|0=1　1|1=1　（或：有真就真，都假才假）
0^0=0　0^1=1　1^0=1　1^1=0　（异或：相等为假，不等为真）
~0=1　~1=0

提 示

> 在对十进制进行位运算时要先转为二进制，然后按上述规则进行计算。

5. 逻辑运算符

逻辑运算通常用来测试真假值。逻辑运算经常用在条件判断和循环处理中，用来判断条件是否满足和是否该跳出循环或继续执行循环。逻辑运算符的具体含义如表 4-5 所示。

表 4-5 逻辑运算符及其意义

<table>
<tr><th>符号</th><th>意义</th></tr>
<tr><td>&&(and)</td><td>逻辑与</td></tr>
<tr><td>||(or)</td><td>逻辑或</td></tr>
<tr><td>xor</td><td>逻辑异或</td></tr>
<tr><td>!</td><td>逻辑非</td></tr>
</table>

逻辑运算的真值表如表 4-6 所示。

表 4-6 逻辑运算的真值表

<table>
<tr><th>$x</th><th>$y</th><th>$x && $y</th><th>$x || $y</th><th>$x xor $y</th><th>!$x</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
</table>

提 示

不要将逻辑与、或、非（&&、||、!）与按位与、或、取反（&、|、~）相混淆。

6. 比较运算符

比较运算符和逻辑运算的用法差不多，通过比较大小来测试真假值，经常用在条件判断和循环处理中，用来判断条件是否满足和是否该跳出循环或继续执行循环。比较运算符的具体含义如表 4-7 所示。

表 4-7 比较运算符及其意义

符号	意义
<	小于
>	大于
<=	小于或等于
>=	大于或等于
==	等于（不包括类型）
===	完全相等（包括类型）
!=	不等于

提 示

“===”为 PHP 4 中新增的比较运算符，用于类型判断例如：“3”===3 将判为假。其他的比较运算符和 C 语言的基本一致，这里就不再多说了。

7. 其他运算符

除了上述的运算符号之外，还有一些难以归类的、用于其他用途的运算符，其具体含义如表 4-8 所示。

表 4-8　其他运算符及其意义

符号	意义
$	用于定义变量
&	变量的地址（加在变量前引用变量）
@	屏蔽错误信息（加在函数前）
->	引用对象的方法或者属性
=>	用于给数组元素赋值
? :	三目运算符

其中比较特殊的是三目运算符“?: ”，例如：

```
(expr1)?(expr2):(expr3);
```

表示如果 expr1 的运算结果为 true，则执行 expr2，否则执行 expr3。实际上它有点类以 if…else 循环，但可以让程序较精简有效率。

此外，还有用于新对象的定义符 new、用于数组下标引用的方括号“[]”、表示结合性的大括号“{}”等。

提 示

> PHP 中的运算符是十分丰富的，而且使用起来也很灵活，希望大家多多上机实践练习哦！

4.3.2　运算符优先级与结合性

运算符优先级决定了两个表达式绑定得有多“紧密”。例如，表达式 1 + 5 * 3 的结果是 16 而不是 18 是因为乘号（“*”）的优先级比加号（“+”）高。必要时可以用括号来强制改变优先级。例如：(1 + 5)* 3 的值为 18。如果运算符优先级相同，则使用从左到右的左联顺序。

表 4-9 从高到低列出了运算符的优先级。同一行中的运算符具有相同优先级，此时它们的结合方向决定求值顺序。

表 4-9　运算符优先级与结合性列表

优先级	结合方向	运算符	附加信息
1（最高）	非结合	new	new
2	左	[	array()
3	非结合	++ --	递增 / 递减运算符

（续表）

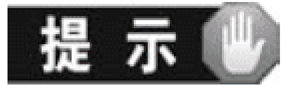
优先级	结合方向	运算符	附加信息
4	非结合	! ~ -(int)(float)(string)(array)(object)@	类型
5	左	* / %	算数运算符
6	左	+ - .	算数运算符和字符串运算符
7	左	<< >>	位运算符
8	非结合	< <= > >=	逻辑运算符
9	非结合	== != === !==	逻辑运算符
10	左	&	位运算符和引用
11	左	^	位运算符
12	左	|	位运算符
13	左	&&	逻辑运算符
14	左	||	逻辑运算符
15	左	? :	三元运算符
16	右	= += -= *= /= .= %= &= |= ^= <<= >>=	赋值运算符
17	左	and	逻辑运算符
18	左	xor	逻辑运算符
19	左	or	逻辑运算符
20	左	,	多处用到

4.3.3 PHP 的表达式

表达式，就是由操作数、操作符以及括号等所组成的合法序列。简单地说，PHP 中的常量或变量通过运算符连接后就形成了表达式，例如“$a=1”为一个表达式。表达式也有值，如上面的这个表达式的值就是 1。

根据表达式中运算符类型的不同，又可以把表达式分成：算术表达式、字符串表达式、赋值表达式、位运算表达式、逻辑表达式、比较表达式、其他表达式等。

最常见的表达式是比较表达式和逻辑表达式，这种表达式的值只能是真或假，在下面一节的流程控制中，会大量使用这两种表达式。



“$a=1”为表达式，“$a=1;”则为一条语句。一定要分清楚哦！

4.4 PHP 的流程控制

在任何一种编程语言中，所有的程序都由 3 种基本的结构组成，即：顺序结构、分支结构、循环结构。

程序是由若干的语句组成的，如果在程序中的语句的执行顺序是从上到下依次逐句执行的，那么这个程序的结构就是顺序结构，在这种结构中没有分支和反复，这也是最简单、最常见的流程结构，这里就不再多讨论。

PHP 中提供了 4 条流程控制语句来实现分支结构和循环结构，即：

①if … else …　　条件语句
②switch　　分支选择语句
③do … while … 循环语句
④for　　循环语句

此外 PHP 还提供了 break 语句和 continue 语句，用以跳出分支结构或循环结构，下面就将这些语句逐一进行详细的介绍。

4.4.1　if…else… 语句

If…else…语句共有三种基本结构，此外每种基本结构还可以嵌套另外两种结构，而且嵌套的层次也可以不止是一层。

1. 只有 if 的语句

这种结构可以当作单纯的判断，可解释成“若某条件成立则去做什么事情”。语句如下：

```
if(expr){
            statement
}
```

其中的 expr 为判断的条件表达式，通常都是由比较表达式或逻辑表达式来充当，而 statement 为符合条件的执行语句部分程序，若程序只有一行，可以省略大括号 {}。如果 expr 为真，则执行 statement 语句或语句体。

2. if…else…语句

这种结构可解释成“若某条件成立则去做什么事情，否则去做什么事情”。其语句如下：

```
if(expr){
            statement1
} else {
            statement2
}
```

如果 expr 为真，则执行 statement1 语句或语句体，否则执行 statement2 语句或语句体。

3. else if 语句

前面的两种分支结构只能实现二路分支，用包含 else if 的语句则可以实现多路分支，其语句如下：

```
if(expr1){
            statement1
}
else if (expr2) {
            statement2
}
        else if …
        else {
statementn
}
```

当 expr1 为真时，执行 statement1 语句或语句体，否则转入 expr2 的判断，如果 expr2 为真，则执行 statement2 语句或语句体，依此类推，如果所有的 expr 表达式都不为真，则执行 statementn 语句或语句体。

现就 if…else…语句举例如下：

程序清单　文档 4-15.php

```
<!--文件 4-15.php: if … else … 的应用-->
<HTML>
 <HEAD>
     <TITLE>if … else … 的应用</TITLE>
 </HEAD>
 <BODY>
     <?php
      //本程序测试时，请更改测试服务器的系统时间查看效果
      echo "今天是: ".date("D")."<br>";
     if(date("D")=="Sat")
          echo "周末了，我们要去狂欢。<br>";
       if(date("D")=="Sat"){
           echo "再次声明，周末了，我们要去狂欢，好高兴! <br>";
           }else{
           echo "今天不是周末，我们要学习 PHP 5，多么有意义啊! ^_^!!!<br>";
           }
       if(date("D")=="Sun"){
           echo "今天是星期天。";
           }
         else if(date("D")=="Mon"){
              echo "今天是星期一。";
              }
         else if(date("D")=="Tue"){
              echo "今天是星期二。";
              }
         else if(date("D")=="Wed"){
```

```
                echo "今天是星期三。";
                }
            else if(date("D")=="Thu"){
               echo "今天是星期四。";
                }
            else if(date("D")=="Fri"){
                echo "今天是星期五。";
                }
            else{
                echo "今天是星期六（只可能是星期六了^_^)。";
                }
        ?>
        </BODY>
</HTML>
```

上例程序中的 date()函数是将服务器的时间格式化函数，date("D")返回服务器时间的星期时间中的英文的前三个字符。当系统时间不为星期六时，其运行效果如图 4-15 所示。

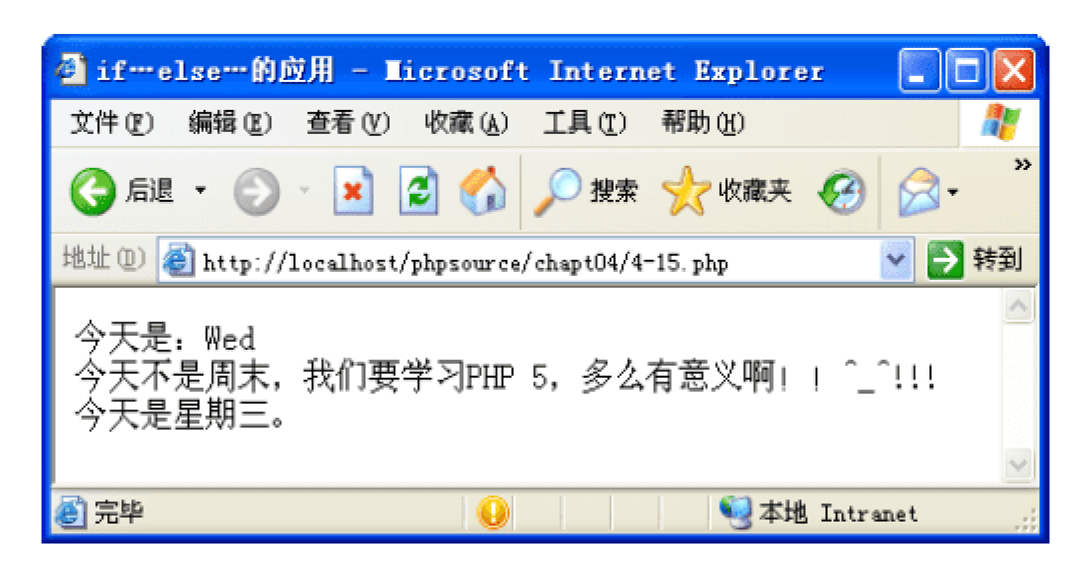


图 4-15　4-15.php 程序的运行结果

在上述 3 种基本结构中，如果在 statement 语句体中还有 if…else…语句，就构成嵌套的 if…else…语句。

提 示

> 在使用嵌套的 if…else…语句时，一定要注意 else 和 if 的匹配、只有 else 语句没有 if 的语句是不存在的。

4.4.2　switch 语句

嵌套的 if…else…语句可以处理多分支流程情况，但使用起来比较繁琐而且分析也不太清晰，为此 PHP 中又引进了 switch 语句。其语句如下：

```
switch(expr){
            case expr1:
            statement1;
            break;
```

```
            case expr2:
            statement2;
            break;
              ⋮
            default:
            statementN;
            break;
}
```

其中的 expr 为条件，通常为变量名称。而 case 后的 exprN，通常表示变量值。冒号后则为符合该条件要执行的部分。一定要注意 break 为跳离循环，千万不能省略不写。具体使用请见下例：

程序清单　文档 4-16.php

```
<!--文件 4-16.php：switch 的应用-->
<HTML>
 <HEAD>
     <TITLE>switch 的应用</TITLE>
 </HEAD>
 <BODY>
 <?php
      //本程序测试时，请更改测试服务器的系统时间查看效果
     switch(date("D")){
        case "Mon":
              echo "今天星期一";
              break;
         case "Tue":
              echo "今天星期二";
              break;
         case "Wed":
              echo "今天星期三";
              break;
         case "Thu":
             echo "今天星期四";
              break;
         case "Fri":
              echo "今天星期五";
              break;
         default:
              echo "今天放假";
              break;
      }
 ?>
```

```
30: </BODY>
31: </HTML>
```

其运行效果如图 4-16 所示。

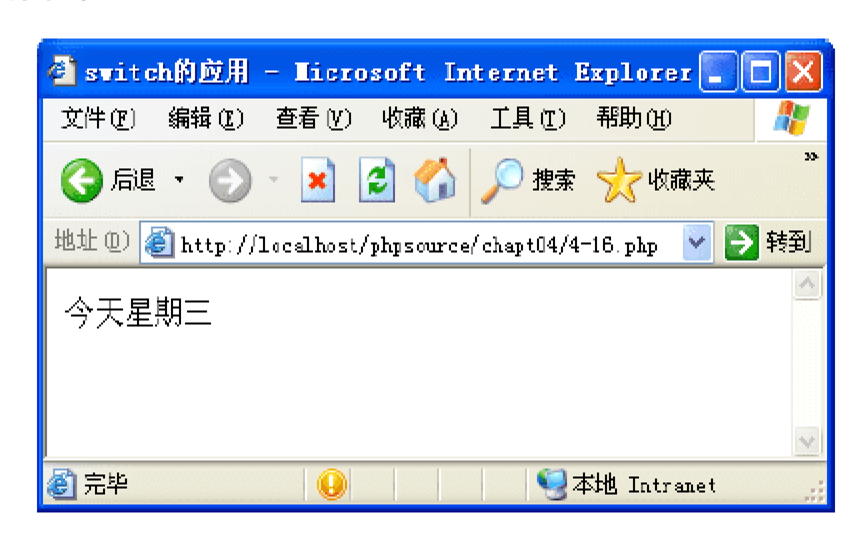


图 4-16　4-16.php 程序的运行结果

通过比较，不难发现例 4-16 要比例 4-15 简单些。

提 示

> 在设计 switch 语句时，要将出现机率最大的条件放在最前面，最少出现的条件放在最后面，可以提高程序的执行效率。例 4-16 由于每天出现的机率相同，所以不用注意条件的顺序。

4.4.3　do…while…语句

在 PHP 中，do…while…循环语句有两种结构，一种只有 while…部分，另一种是 do…while…两部分都有。

1. 只有 while…部分的语句

其语句如下：

```
while(expr){
            statement
}
```

2. do…while…两部分都有的语句

其语句如下：

```
do {
            statement
} while(expr);
```

其中的 expr 为判断的条件，通常都是用逻辑表达式或比较表达式充当判断的条件。而 statement 为符合条件的执行程序部分，若程序只有一行，可以省略大括号{}。

二者的区别在于：前者是先判断再执行，后者是先执行再判断。对于 expr 开始为真的情况，二者是没有区别的。如果 expr 开始为假，则前者不执行任何语句就跳出循环，而后者仍要执行一次循环才能跳出循环。

具体使用如下例：

程序清单　文档 4-17.php

```
<!--文件 4-17.php：do…while…的应用-->
<HTML>
 <HEAD>
     <TITLE>do…while…的应用</TITLE>
 </HEAD>
 <BODY>
 <?php
      //本程序测试时，请更改两处$i 的初始值为 11
      //观看效果，分析区别
      $i=1;
       while($i<=10){
            print $i++;
            echo ". 以后不敢了<br>\n";
       }
       echo "第二次…<br>";
       $i=1;
       do{
            print $i++;
            echo ". 以后不敢了<br>\n";
      } while($i<=10)
     ?>
     </BODY>
</HTML>
```

其运行效果如图 4-17 所示。

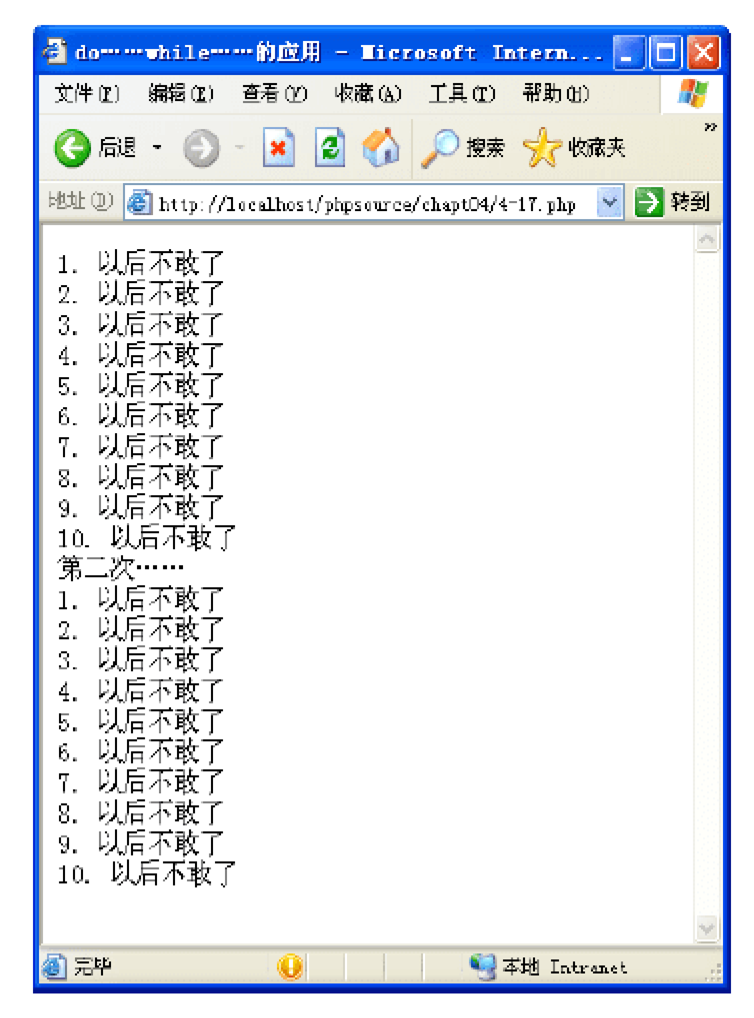


图 4-17　4-17.php 程序的运行结果

4.4.4　for 语句

for 语句仅有单纯的一种，没有其他变化，但同时它也是最复杂、功能最强大的循环，任何 while 循环和 do…while…循环都可以用 for 循环代替，其语句如下：

```
for(expr1; expr2; expr3){
    statement
}
```

其中 expr1 为条件的初始值，expr2 为判断的终值，通常都是用比较表达式或逻辑表达式充当判断的条件，expr3 为执行 statement 后要执行的部分，即循环步长，用来改变条件，供下次的循环判断，如加 1、减 1 等等。而 statement 为符合条件后执行的语句或语句体，若 statement 只有一条语句组成，则可以省略大括号 {}。

例 4-17 程序的主体部分可以用 for 循环改写如下：

```
<?php
for($i=1;$i<=10;$i++){
   echo $i.". 以后不敢了<br>";
}
?>
```

当然还可以改写成如下 3 种形式：

```
形式 1：<?php
       $i=1;
                for(;$i<=10;$i++){
                echo $i.". 以后不敢了<br>";
             }
       ?>

       <?php
形式 2：for($i=1;;$i++){
                     if($i>10) break;    //循环标记结束，跳出循环
                     echo $i.". 以后不敢了<br>";
                 }
        ?>

形式 3：<?php
       for($i=1;$i<=10;){
                    echo $i.". 以后不敢了<br>";
                    $i++;
                 }
       ?>
```

从上例中，可以明显地看到，用 for 和用 while 的不同。实际应用上，若循环有初始值，且都要累加（或累减），则使用 for 循环比用 while 循环好。

4.4.5 break 和 continue 语句

在 switch 和 for 语句的举例中都用到了 break 语句，它的作用就是跳出整个的 switch 或 for 语句体，执行循环体后面的语句。而 continue 经常用在 for 或 do…while…循环语句中，表示跳出本次循环，继续进入下一次的循环。这也是 break 和 continue 的主要区别，现举例如下：

程序清单 文档 4-18.php

```
<!--文件 4-18.php: break/continue 的应用-->
<HTML>
 <HEAD>
     <TITLE>break/continue 的应用</TITLE>
 </HEAD>
 <BODY>
         <?php
             echo "使用 break 的输出效果: <br>";
             $i=0;
             while($i<10){
             if($i%2==1){
                 break;
             }
         echo $i;
         $i++;
             }
             echo "<br>使用 contine 的输出效果: <br>";
             for($i=0;$i<10;$i++){
                 if($i%2){
                     continue;
                 }
             echo $i." ";
             }
         ?>
     </BODY>
 </HTML>
```

其运行结果如图 4-18 所示。

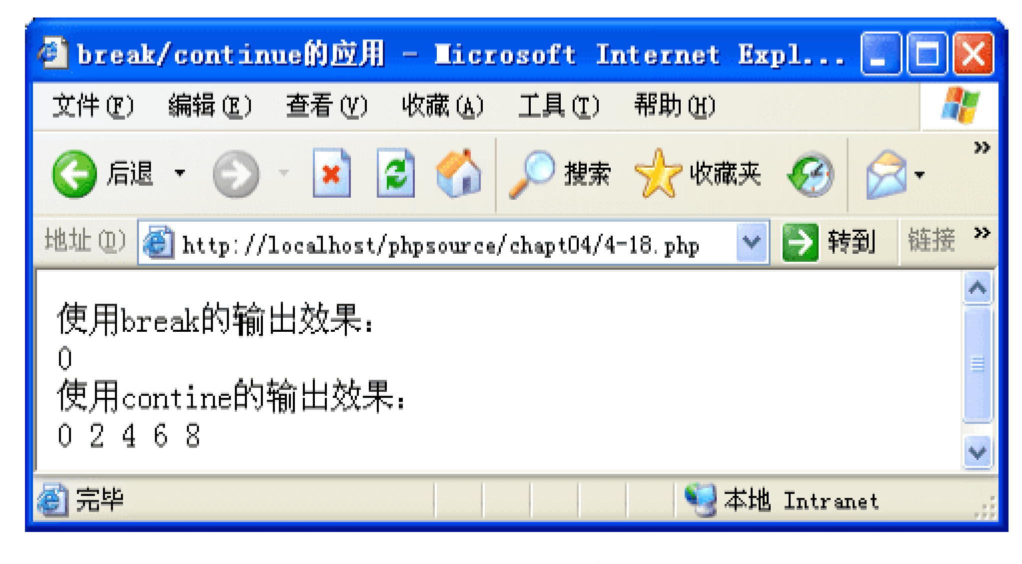


图 4-18 4-18.php 程序的运行结果

4.5　PHP 的系统函数与自定义函数

在程序的编写过程中往往会有一些要反复用到的功能模块，如果每次都要重复编写这些代码，不仅浪费时间，而且还会使程序变得冗长、可读性差，维护起来也很不方便。PHP 5 允许程序设计者将常用的流程或者变量等元件，组织成一个固定的格式。也就是说用户可以自行组合函数或者是类。这样在编写好函数或类之后，使用时就不必关心其中的细节，拿过来就可以用，要做功能修改时，只需修改函数中的内容即可。下面就来看一下自定义函数的定义与使用。

4.5.1　函数的定义和调用

PHP 5 中的函数和 C 语言中的一样。函数定义的语句为：

```
function 函数名(形式参数列表){
        函数体;
        return 返回值;
}
```

PHP 5 中的函数有无返回值都允许。而且，在函数的名称上，PHP 对于大小写的管制比较松散。可以在定义函数时写成大写的名字，而在使用时使用小写的名字。不过，PHP 5 对用户自定义函数名也是有一些要求的，具体如下：

- 不能与 PHP 的内部函数同名；
- 不能与 PHP 的关键字重名；
- 不能以数字或下划线开头；
- 不能包含点号“.”和中文字符。

函数体是实现函数功能的语句体，与 for 循环等不一样，函数体中即使只有一条语句，外面的大括号也不能省略。

函数调用的语句为：

```
函数名（实际参数列表);
```

实际参数列表要与形式参数列表相对应（有默认参数时，实际参数还要与默认参数对应)。如果实际参数比形式参数多，多余的参数会被自动舍弃；如果实际参数比形式参数少，实际参数会被一一填入形式参数中，不足部分以空参数代替。实际参数和形式参数之间的传递机制会在 4.5.2 节中讲述。

如果函数有返回值，还可以利用函数调用为变量赋值，其语句为：

```
变量名=函数名（实际参数列表);
```

现就函数的定义和调用举例如下：

程序清单 文档 4-19.php

```
<!--文件 4-19.php: 函数的定义和调用=>求阶乘-->
<HTML>
 <HEAD>
     <TITLE>函数的定义和调用=>求阶乘</TITLE>
 </HEAD>
 <BODY>
     <?php
     function factorial($n)
     {
         $result=1;
         for($i=2;$i<=$n;$i++)
             $result*= $i;
         return $result;
     }
     for($i=1;$i<=6;$i++)
     {
         $num=factorial($i);
         echo $i."!=".$num."<br>";
     }
     ?>
     </BODY>
</HTML>
```

其运行效果如图 4-19 所示。

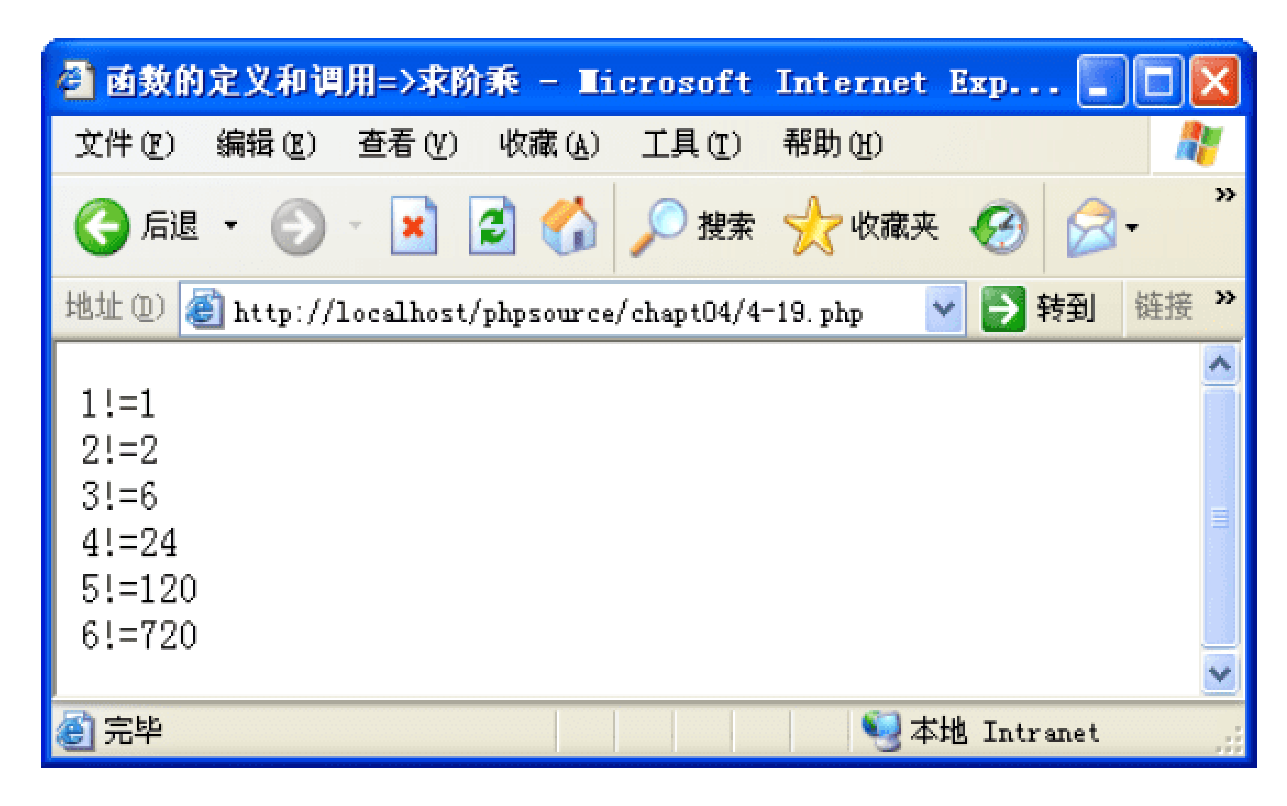


图 4-19 4-19.php 程序的运行结果

4.5.2 函数的参数传递

在使用函数时，要填入与函数形式参数个数相同的实际参数（有默认参数的除外），在程序运行过程中，实际参数就会将参数值传递给相应的形式参数，然后在函数中实现对数据的处理和返回。在由实际参数向形式参数传递的过程中，共有值传递、引用传递、默认

参数 3 种机制。

1. 值传递

值传递，就是将实际参数的值复制到形式参数中。然后由形式参数参与函数内部的计算，函数调用结束后，实际参数的值不会发生改变。

用这种方式调用的函数一般都有返回值或输出值，否则函数调用实际上什么也没有做。例 4-19 就是使用的这种传递方式。初学者一般常使用这种传递方式。

2. 引用传递

如果要实现形式参数改变时实际参数也发生相应的改变，就要使用引用传递的方式。参数的引用传递有两种方法：

①在函数定义的时候，在形式参数前面加上“&”符号，例如：

```
function fun(&$var1){……}
```

②在函数调用的时候，在实际参数前面加上“&”符号，例如：

```
function fun($var1){……}
fun(&$var2)
```

如果形式参数$var1 的值在函数中发生改变，实际参数$var2 的值也会发生相应的改变。

3. 默认参数

PHP 5 支持有默认值的参数，即在函数定义时可以为一个或多个形式参数指定默认值。

现就函数的参数传递的 3 种方式举例如下：

程序清单　文档 4-20.php

```
<!--文件 4-20.php：函数参数的传递-->
<HTML>
     <HEAD>
         <TITLE>函数参数的传递</TITLE>
     </HEAD>
     <BODY>
     <?php
     function myfun1($var1)
     {   //值传递测试函数
          $var1=88;
     }
       function myfun2(&$var1)
     {   //第一种引用传递测试函数
          $var1=88;
     }
       function myfun3($var1)
     {   //第二种引用传递测试函数
          $var1=99;
```

```
    }
      function myfun4($string,$color="red")
        {   //默认参数传递测试函数
          echo "<font color=".$color.">".$string."</font>";
        }
       $var1=66;
       $string="这是红色字体！";
       echo "初始值:\$var=".$var1."<br>";
       myfun1($var1);
       echo "值传递结束后:\$var=".$var1."<br>";
       myfun2($var1);
       echo "第一种引用传递结束后:\$var=".$var1."<br>";
       myfun3(&$var1);
       echo "第二种引用传递结束后:\$var=".$var1."<br>";
       echo "默认参数的效果为:";
       myfun4($string);
        ?>
    </BODY>
</HTML>
```

其运行效果如图 4-20 所示。

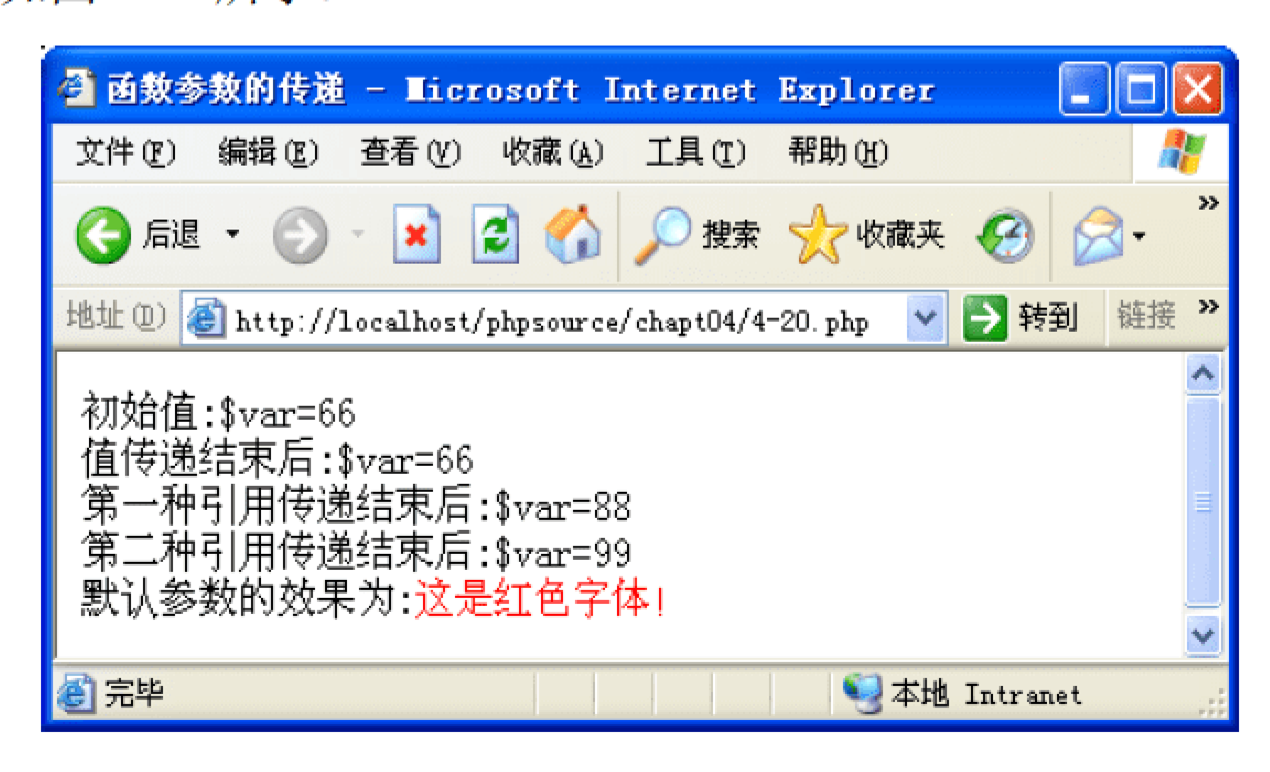


图 4-20 4-20.php 程序的运行结果

4.5.3 用函数的同名变量实现可变函数

PHP 5 中提供了可变函数的机制，可以方便地利用变量实现对函数的调用。使用的基本格式为：

```
Function fun(){……}
$var="fun";
$var();
```

其中调用$var()就相当于调用函数 fun()，$var()为可变函数。可变函数与普通函数调用时的最大区别就在于可变函数前面有“$”，有此符号，系统就会认为是可变函数。

提　示

> 如果写成$var= “fun()” ,则 “$var” 只是一个普通的字符串而已，再调用语句 “$var;” 不会实现 fun()函数的功能。

现举例如下：

程序清单　文档 4-21.php

```
<!--文件 4-21.php：函数的同名变量实现可变函数-->
<HTML>
 <HEAD>
     <TITLE>函数的同名变量实现可变函数</TITLE>
 </HEAD>
 <BODY>
 <?php
     function myfun()
     {
           return "函数的同名变量实现可变函数成功！";
     }
       $var="myfun";
       echo $var();
       echo "<br>";
       $var="myfun()";
       echo $var;
 ?>
 </BODY>
</HTML>
```

其运行结果如图 4-21 所示。

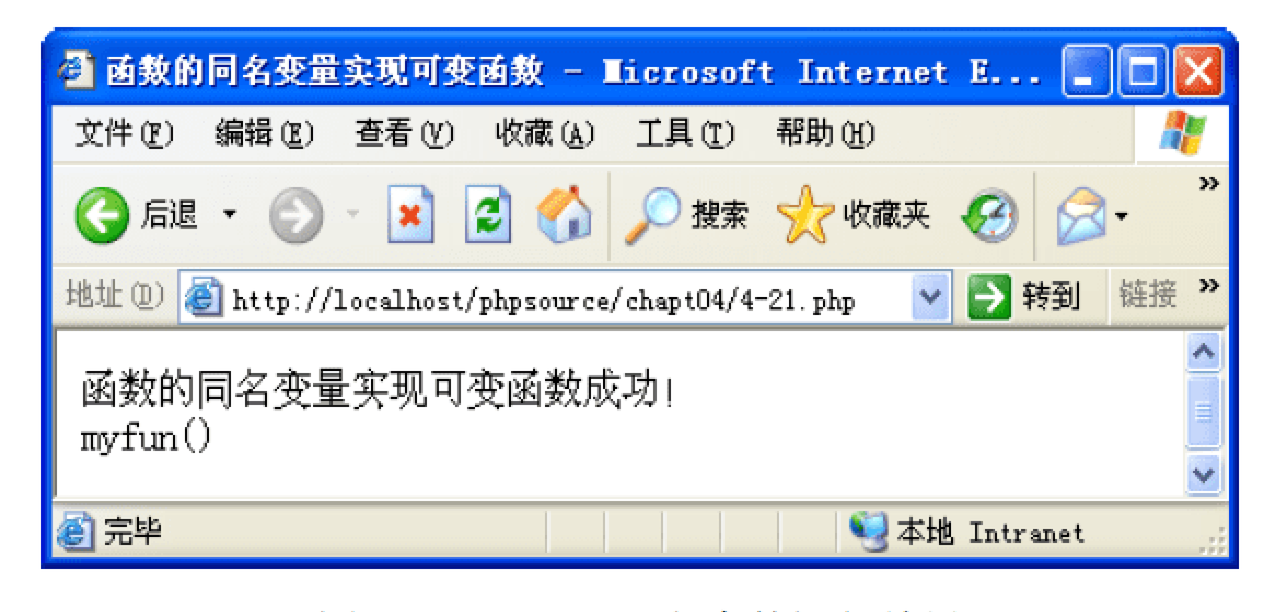


图 4-21　4-21.php 程序的运行结果

4.5.4　用函数调用实现递归与嵌套

PHP 5 中的函数支持递归与嵌套。所谓递归就是一个函数在自己的函数体内调用自身。所谓嵌套，就是一个函数在其函数体内调用其他函数。

下面就用函数递归的方式实现程序 4-19 的功能。

程序清单　文档 4-22.php

```
<!--文件 4-22.php：函数递归实现阶乘运算-->
<HTML>
 <HEAD>
     <TITLE>函数递归实现阶乘运算</TITLE>
 </HEAD>
 <BODY>
 <?php
     function factorial($n)
     {
         if($n==1)
             return 1;
         return $n*factorial($n-1);
     }
      for($i=1;$i<=6;$i++)
     {
         $num=factorial($i);
         echo $i."!=".$num."<br>";
     }
  ?>
 </BODY>
 </HTML>
```

其运行结果如图 4-22 所示。

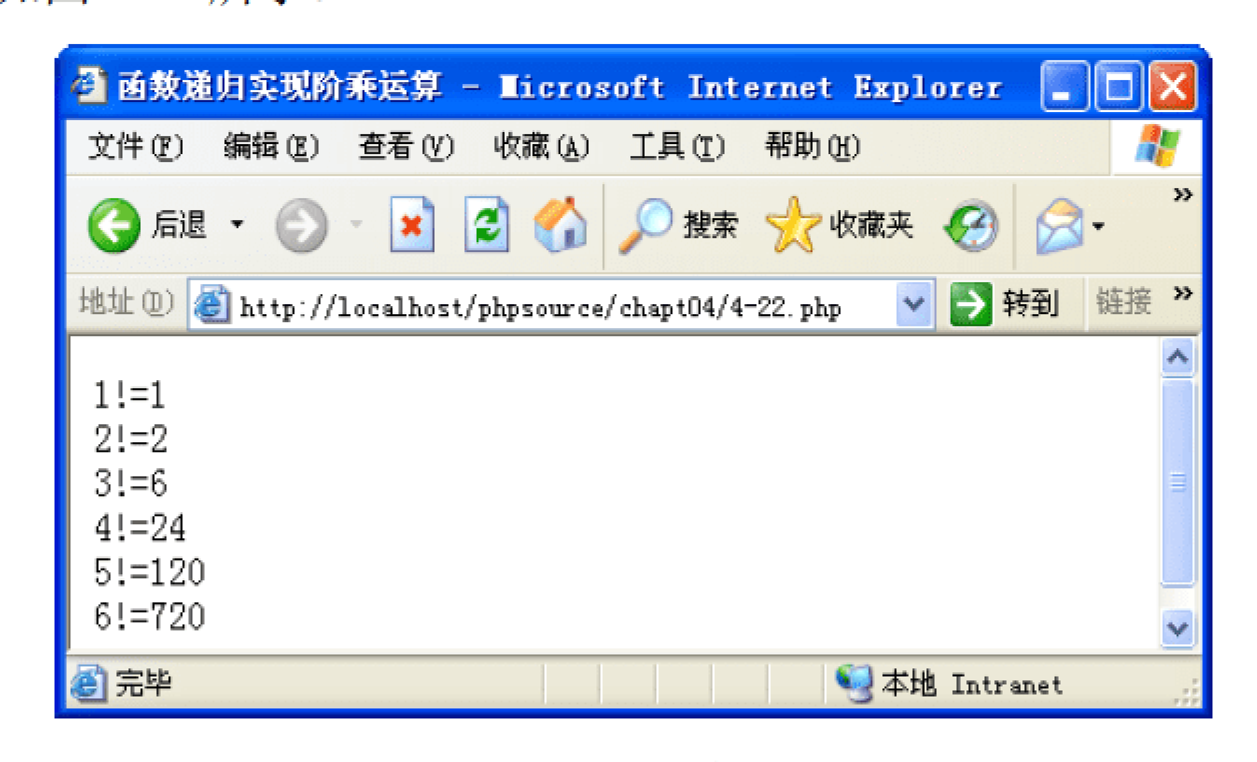


图 4-22　4-22.php 程序的运行结果

通过分析，可以看出，使用递归可以使程序简化，但由于递归的实现是由栈来完成的，每一次函数调用都会用栈来保存信息，所以使用递归会耗用较多的内存空间，而且递归的层数越多，资源耗用得越大。一些比较典型的编程问题，如“八皇后问题”、“骑士遍历问题”等都可以使用 PHP 中的递归函数思想给出漂亮的解法，下面再用递归的思想输出斐波那契数列。

程序清单　文档 4-23.php

```
<!--文件 4-23.php：使用递归求解斐波那契数列-->
<HTML>
 <HEAD>
     <TITLE>使用递归求解斐波那契数列</TITLE>
 </HEAD>
 <BODY>
 <?php
     function fib($n)
     {
         if($n<0)
             return 0;
         if($n<=2)
             return 1;
         return fib($n-1)+fib($n-2);
     }
       echo "斐波那契数列的前 10 项：";
     for($i=1;$i<=10;$i++)
     {
         $Var=fib($i);
         echo $Var." ";
         }
 ?>
 </BODY>
</HTML>
```

其运行效果如图 4-23 所示。

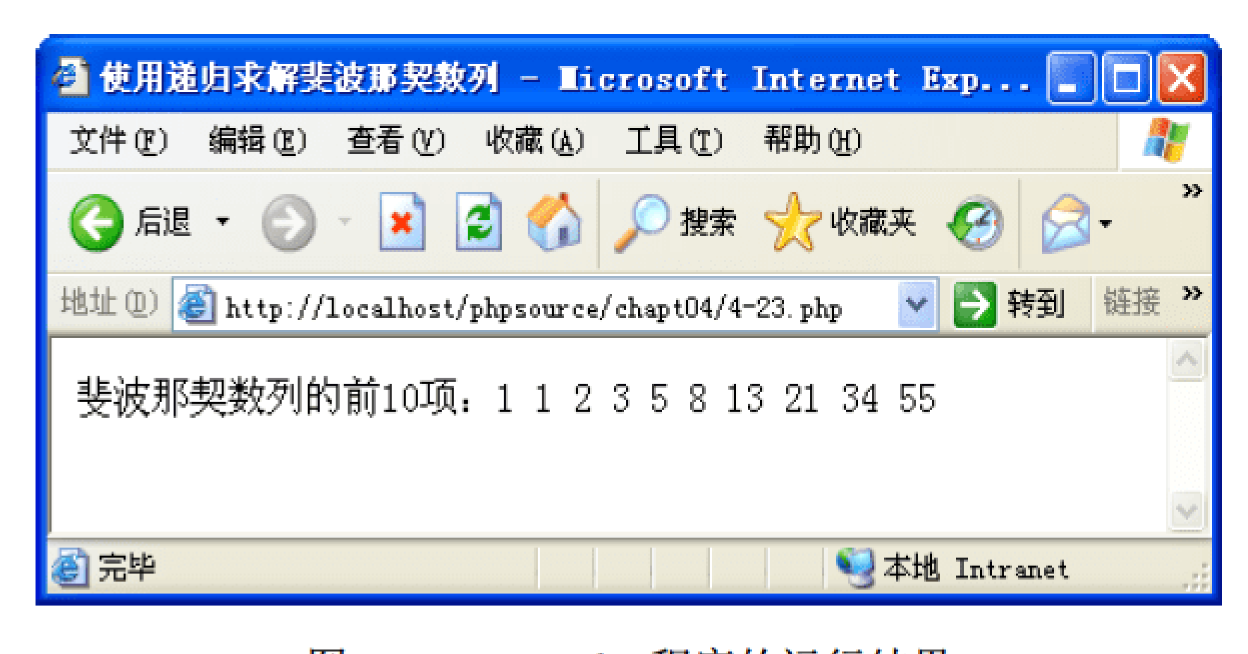


图 4-23　4-23.php 程序的运行结果

下面再举一个使用函数嵌套求闰年的例子，闰年的计算方法为：公元纪年的年数可以被四整除，即为闰年；被 100 整除而不能被 400 整除为平年；被 100 整除也可被 400 整除的为闰年。如 2000 年是闰年，而 1900 年不是。

程序清单　文档 4-24.php

```
<!--文件 4-24.php：函数嵌套求闰年-->
<HTML>
  <HEAD>
      <TITLE>函数嵌套求闰年</TITLE>
  </HEAD>
  <BODY>
  <?php
      function div4($a) //判断是否能为 4 整除
      {
          if($a%4==0)
              return true;
          else
              return false;
      }
      function div100($a) //判断是否能为 100 整除
      {
          if($a%100==0)
              return true;
          else
              return false;
      }
      function div400($a) //判断是否能为 400 整除
      {
          if($a%400==0)
              return true;
          else
              return false;
      }

      function isleapyear($a)
      {
          if(!div4($a))
              echo "$a 年不是闰年。<br>";
          elseif(div100($a)&&!div400($a))
              echo "$a 年不是闰年。<br>";
          else
              echo "$a 年是闰年。<br>";
      }
      isleapyear(1900);
      isleapyear(1996);
      isleapyear(1999);
```

```
    isleapyear(2000);
?>
</BODY>
</HTML>
```

其运行效果如图 4-24 所示。



图 4-24　4-24.php 程序的运行结果

4.5.5　变量在函数中的使用

PHP 5 的变量就使用范围的不同可以分为若干种类，比如局部变量、全局变量、静态变量、动态变量、引用变量、预定义变量、外部变量等，下面就前 5 种变量的进行讲解。

1. 局部变量和全局变量

PHP 中的变量是有作用范围的，一般情况下，变量的作用域是包含变量的 PHP 程序块，也就是“<?php”、“?>”之间的代码或者“{”、“}”之间的代码。如果其中还有 include()或 require()函数包含的文件，变量的作用域也可以扩展到这些文件里。

在主程序中定义的变量只在主程序中有效，在函数中定义的变量只在函数中有效，在函数之外就无效了。

如果要在函数中引用主程序中的变量的值，使用 PHP 定义的$GLOBALS 数组，$GLOBALS 数组是一个以变量名为下标的数组，可以直接调用。

还有一种实现跨域调用功能的方法就是使用全局变量，值得注意的是 PHP 中的全局变量并不是对整个程序都可见，在主程序中可以访问函数中的全局变量，但是在函数中并不能访问主程序的全局变量。其定义的语句为：

```
global  $变量名;
```

只要定义了变量为全局变量，以后访问这个变量是就不需要使用$GLOBALS 数组了。下面就全局变量和局部变量举例如下：

程序清单　文档 4-25.php

```
<!--文件 4-25.php：局部变量和全局变量-->
<HTML>
  <HEAD>
```

```
    <TITLE>局部变量和全局变量</TITLE>
</HEAD>
<BODY>
<?php
     $var1=6;  //定义一个局部变量
     echo "定义局部变量\$var=".$var1;
     include("a.php");
     function myfun1(){
         echo "<br>在 myfun1()函数中输出局部变量";
         echo "\$var1=".$var1;
         }
     myfun1();
     function myfun2(){
         echo "<br>在 myfun2()函数中输出局部变量";
         echo "\$var1=".$GLOBALS["var1"];
         }
     myfun2();
     global $var2;
     $var2=8;
     function myfun3(){
         global $var3;
         $var3=9;
         echo "<br>在 myfun3()函数中输出全局变量";
         echo "\$var2=".$var2;
         echo " \$var3=".$var3;
         }
     myfun3();
     echo "<br>在函数外输出全局变量";
     echo "\$var2=".$var2;
     echo " \$var3=".$var3;
?>
</BODY>
</HTML>
```

```
<!--文件 a.php: 文件 4-25.php 所包含的文件-->
<?php
    echo "<br>在 a.php 中输出\$var 的值";
    echo "\$var1=".$var1;
    echo "<br>在 a.php 中的函数中输出\$var 的值";
    function myfun4(){
        echo "\$var1=".$var1;
        }
?>
```

其运行结果如图 4-25 所示。

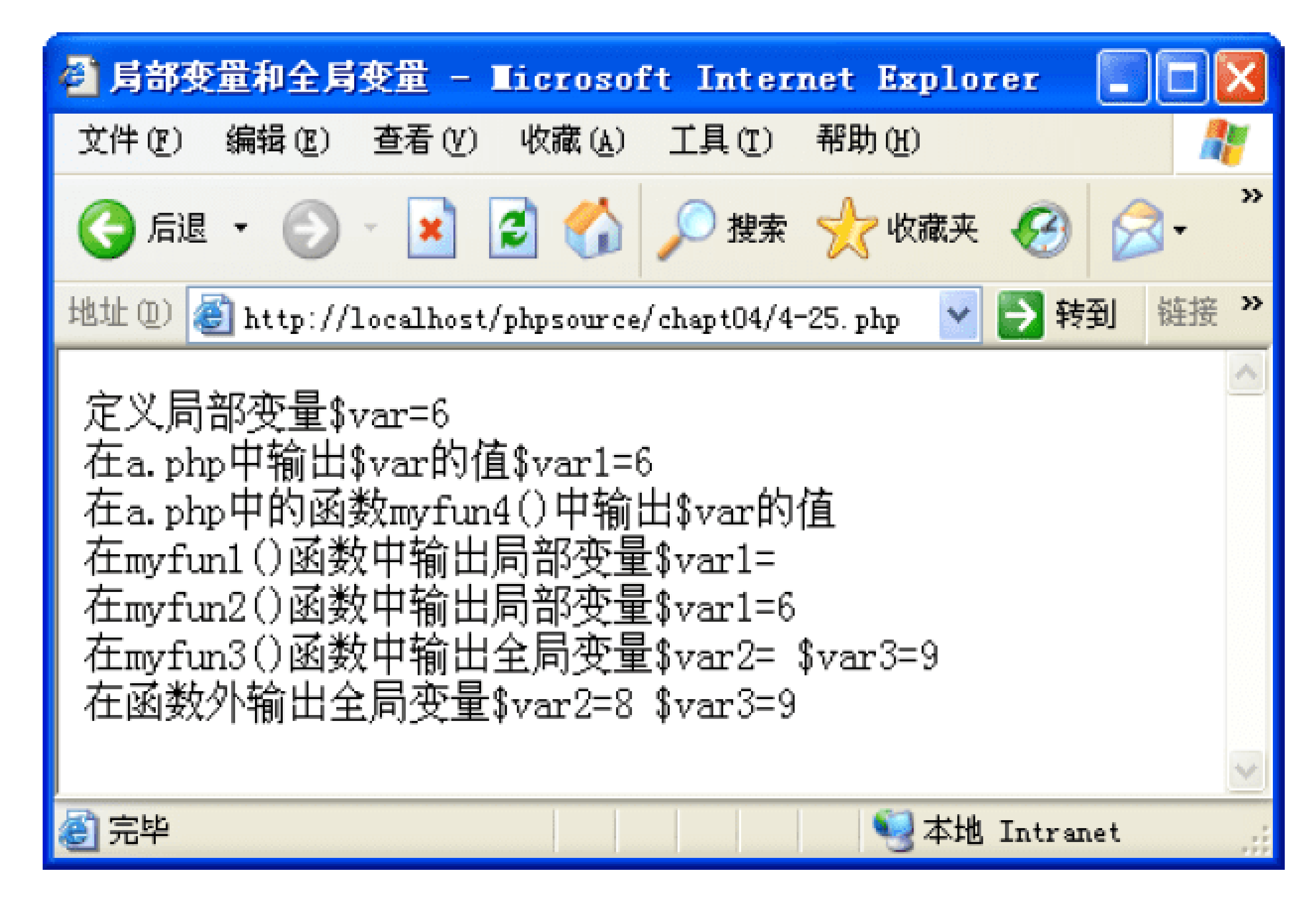


图 4-25　4-25.php 程序的运行结果

提 示

在 PHP 函数中使用$GLOBALS 数组，是不是和 4.5.2 节中讲到的参数传递有异曲同工之效呢？为了防止程序中变量的混乱，建议尽量不要使用全局变量，至少应尽可能少地使用全局变量。

2. 静态变量和动态变量

静态变量只能用于函数范围内，与普通局部变量不同之处在于：当该函数运行结束时，静态变量不会丢失自己的值。静态变量的定义方法为在变量名前冠以 static 关键字。

PHP 中的动态变量允许一个变量的值作为另一个变量的变量名，前提条件是这个值要符合变量命名的规则。现举例如下：

程序清单　文档 4-26.php

```
<!--文件 4-26.php：静态变量与动态变量-->
<HTML>
  <HEAD>
      <TITLE>静态变量与动态变量</TITLE>
  </HEAD>
  <BODY>
  <?php
      function fun1()
      {
          $int1=0;        //定义普通变量
          echo $int1." ";
          $int1++;
      }
      fun1();    //输出 0
```

```
    fun1();    //输出还是 0
    echo "<br>*************<br>";
    function fun2()
    {
        static $int2=0;       //定义静态变量
        echo $int2." ";
        $int2++;
    }
    fun2();    //输出 0
    fun2();    //输出 1
     echo "<br>*************<br>";
     $int=6;
     $var="int";
     echo $$var;   //输出动态变量
     $$var=8;
      echo " ".$int;
?>
</BODY>
</HTML>
```

其运行结果如图 4-26 所示。

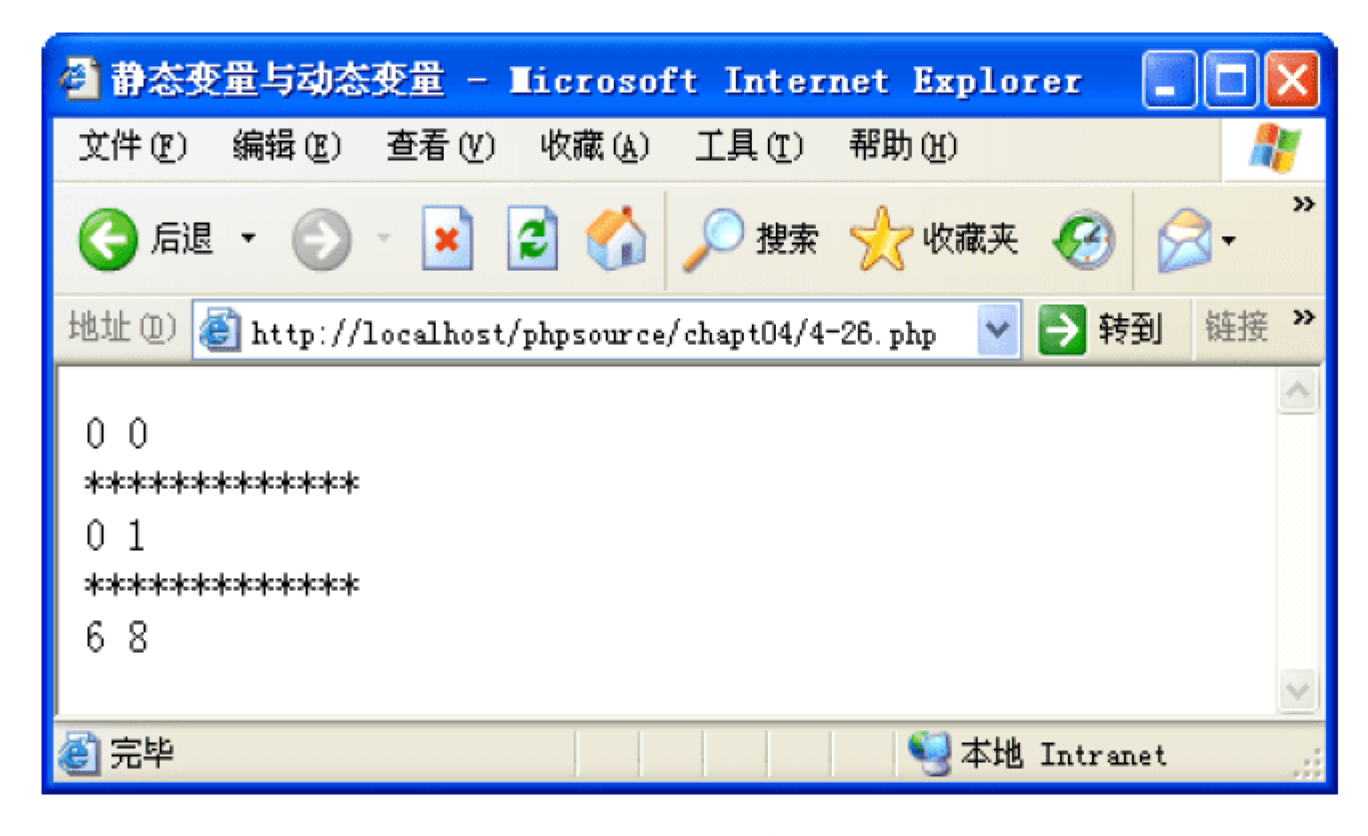


图 4-26　4-26.php 程序的运行结果

通过上例不难看出静态变量与普通变量的区别，在函数 fun2()中静态变量$int2 只被初始化了一次，在该函数退出时$int2 的值不会丢失，所以反复调用函数 fun2()时，$int2 的值会累加。

提 示

> 静态变量虽然在函数结束时值不会丢失，但其作用域只在本函数中，在函数之外仍然无效。动态变量的实现思想是否与 4.5.3 节中用函数的同名变量实现可变函数的思想相吻合呢？

3. 引用变量

PHP 5 中支持引用变量，这是从 C++语言中借用而来的。对一个变量进行引用产生新变量后，新变量可以看作是原变量的一个别名，改变其中任何一个变量的值，另一个变量的值也会随之发生改变。使用引用变量的方法是：赋值时在等号右边变量前加“&”符号，具体实例如下：

程序清单　文档 4-27.php

```
<!--文件 4-27.php：引用变量-->
<HTML>
  <HEAD>
      <TITLE>引用变量</TITLE>
  </HEAD>
  <BODY>
  <?php
       $int1=8;
       $int2=&$int1;
       $int2++;
       echo $int1;
  ?>
  </BODY>
  </HTML>
```

其运行效果如图 4-27 所示。

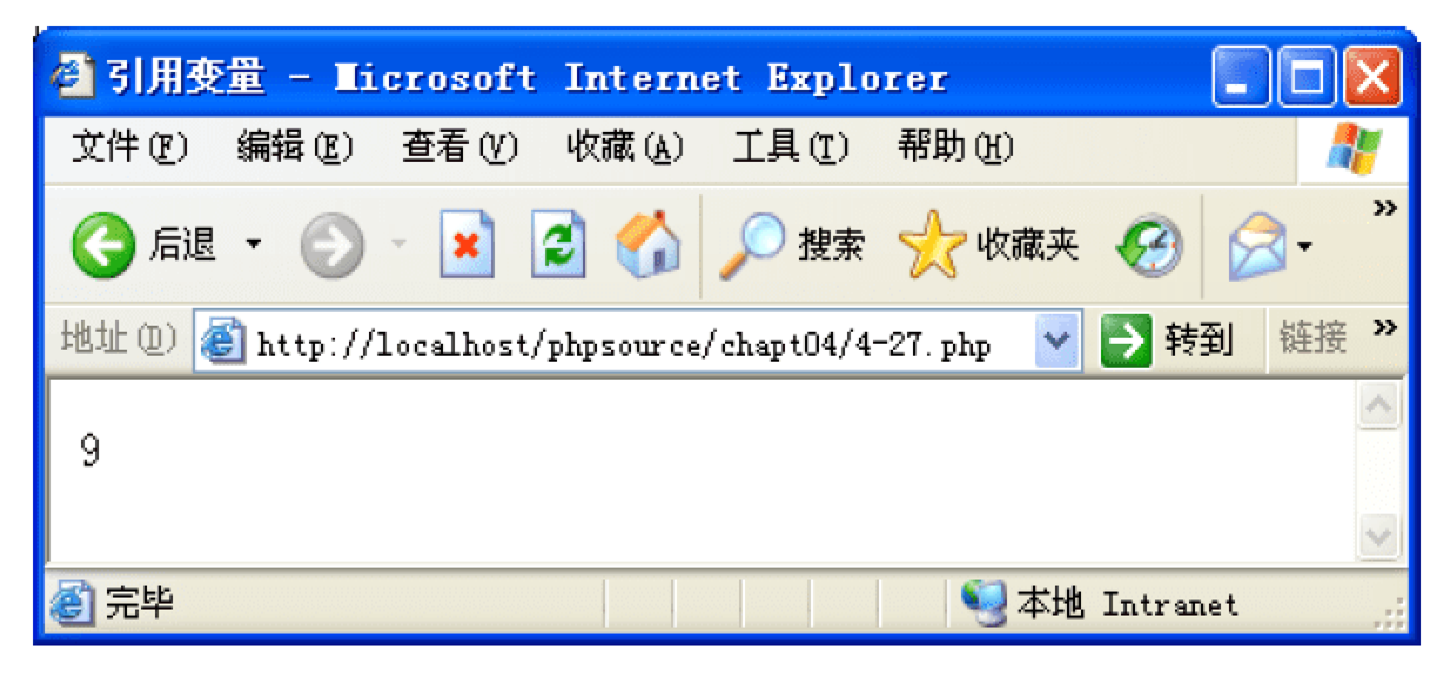


图 4-27　4-27.php 程序的运行结果

引用变量在进行赋值时，系统只是简单将新变量执行原变量的地址，其中无变量复制的操作，所以速度非常快。如果参数是大数组或对象，使用引用变量进行参数传递能节省内存并加快程序运行速度。不过其副作用就是引用值改变，原变量也会发生改变，因此是否使用引用变量进行参数传递要视具体情况而定。

4.6 PHP 5 的类与对象

PHP 5 新的面向对象编程特性显著提升了这个流行语言中的功能层次。

要想成为一名合格的 PHP 程序员，必须掌握类和对象。您肯定知道变量和函数，但类和对象可能就是另一回事。如果您不定义、使用类和对象，您就无法创建完美的系统。即使决定在自己的代码中不使用面向对象的编程，您仍可能需要了解面向对象的编程。比如，如果您要使用第三方库，那么您就必须学会实例化对象和调用方法。下面就来逐步学习类和对象的相关知识。

4.6.1 类和对象的概念

简单地说，类是一个由变量和方法组成的独立程序块或功能模块。这些变量和方法的结合通常能实现单个功能或一组功能。类可以直接用作组织数据和功能的简单方法，就像一组函数和变量一样，但使用类可以忽略函数和变量的存在。类可用于在内存中生成多个实例。这样的实例我们就称作对象。每个对象可以访问一组相同的函数（在面向对象上下文中叫做方法）和变量（叫做特性或实例变量），但每个变量的实际值在每个对象中是不同的。

我们知道，角色可以扮演游戏中的一个单元——比如坦克。类可为坦克设置一组变量：防御和进攻能力、范围、健康状况等等。这个类也可以定义一组函数，其中包括 move()和 attack()函数。当系统包含一个坦克类时，这个类可用于生成数十个或数百个坦克对象，每个对象都潜在地具有自己的健康状况或范围特征。因此，类是用于生成对象的蓝图或模板。

4.6.2 类的定义和对象的创建

理解类和对象最简单的方法应该就是创建一些类和对象。类的定义的语法如下：

```
class       类名(参数类表){
代码块（类变量集与函数集）
}
```

我们可以看出，类是由关键字、类的名称和代码块组成。其中类名可以包含字母、数字和下划线字符的任何组合，但不能以数字打头。

对象创建的方法为：

```
$对象名=new 类名(参数类表);
```

我们必须使用 new 关键字来创建新的对象。这就是告诉 PHP 引擎，您希望实例化一个新对象。然后，返回的对象可以存储在一个变量中以供将来使用。在形式上，实例化对象与调用函数相似。值得注意的是实例化对象可以有参数类表，也可以无参数类表，但如果无参数类表，一定把括号也去掉。例 4-28 中就是无参数类表的情形，后面的例 4-31 就用到了无参数类表的情形。

在类的主体中，可以声明叫做属性的特殊变量。在 PHP 4 中，属性必须用关键字 var 调用。这在 PHP 5 仍是合法的语法，但主要是为了提高 PHP 5 的兼容性。事实上，PHP 5

中属性的可见度是可以设置为 public、private 和 protected。声明为 public 的属性可以在类外部写入和读取，声明为 private 的属性只在对象或类上下文中可见。声明为 protected 的属性只能在当前类及其子类的上下文中可见（在继承部分将会看到这些内容起作用）。可以使用 private 属性来真正锁定类。如果将属性声明为 private 并试图从类范围外部访问它，PHP 引擎将抛出致命错误。

方法，简单地说就是在类中声明的函数。我们来使用对象的某个方法时的语句为：

```
$对象名->方法名(参数类表);
```

下面我们来看一个具体的事例：

程序清单　文档 4-28.php

```
<!--文件 4-28.php：类的定义和对象的创建-->
<HTML>
 <HEAD>
     <TITLE>类的定义和对象的创建</TITLE>
 </HEAD>
 <BODY>
 <?php
        class point
        {
            public $x;      //点 x 的坐标
            //private $x;
            public $y;      //点 x 的坐标
            //private $y;
            function setx($set_x)
            {
                $this->x=$set_x;
                }
            function sety($set_y)
            {
                $this->y=$set_y;
                 }
             function print_point()
             {
                 echo "坐标为：(".$this->x.",".$this->y.") ";
                 }
             }
         if($point1=new point){
             echo "对象\$point1 创建成功！<br>";
             }
        $point1->setx(6);
        //$point1->x=6;
```

```
32:       $point1->sety(8);
33:       //$point1->y=6;
34:       $point1->print_point();
35:
36: ?>
37: </BODY>
38:</HTML>
```

其运行效果如图 4-28 所示。



图 4-28　4-28.php 程序的运行结果

对于上面的实例程序，依次作下面的 3 种修改，并查看结果，不难发现 public 和 private 的区别：

① 去掉第 31、33 行的屏蔽，屏蔽第 30、32 行；
② 去掉第 11、13 行的屏蔽，屏蔽第 10、12 行；
③ 同时执行①和②的操作。

4.6.3　用构造函数实现类的初始化

在日常生活中，我们往往希望类中的一些属性，在声称对象的时候有个初始值，使用类的构造函数就可以达到此目的。

构造函数是类中的一个特殊函数，这个函数的函数名与类名相同，而且在对象生成时会自动调用。下面就对例 4-28 进行改造，改造结果如下：

程序清单　文档 4-29.php

```
<!--文件 4-29.php：构造函数实现类的初始化-->
<HTML>
 <HEAD>
     <TITLE>构造函数实现类的初始化</TITLE>
 </HEAD>
 <BODY>
 <?php
       class point
       {
           public $x;      //点 x 的坐标
```

```
11:            public $y;      //点 x 的坐标
12:            function point() //类的同名函数做构造函数
13:            {
14:                $this->x=6;   //初始化点 x 的值
15:                $this->y=6;   //初始化点 y 的值
16:                }
17:            function setx($set_x)
18:            {
19:                $this->x=$set_x;
20:                }
21:            function sety($set_y)
22:            {
23:                $this->y=$set_y;
24:                }
25:            function print_point()
26:            {
27:                echo "坐标为：(".$this->x.",".$this->y.")";
28:                }
29:            }
30:        $point1=new point;
31:          //$point1->setx(6);
32:        //$point1->sety(8);
33:        $point1->print_point();
34: ?>
35: </BODY>
36:</HTML>
```

其运行结果如图 4-29 所示。

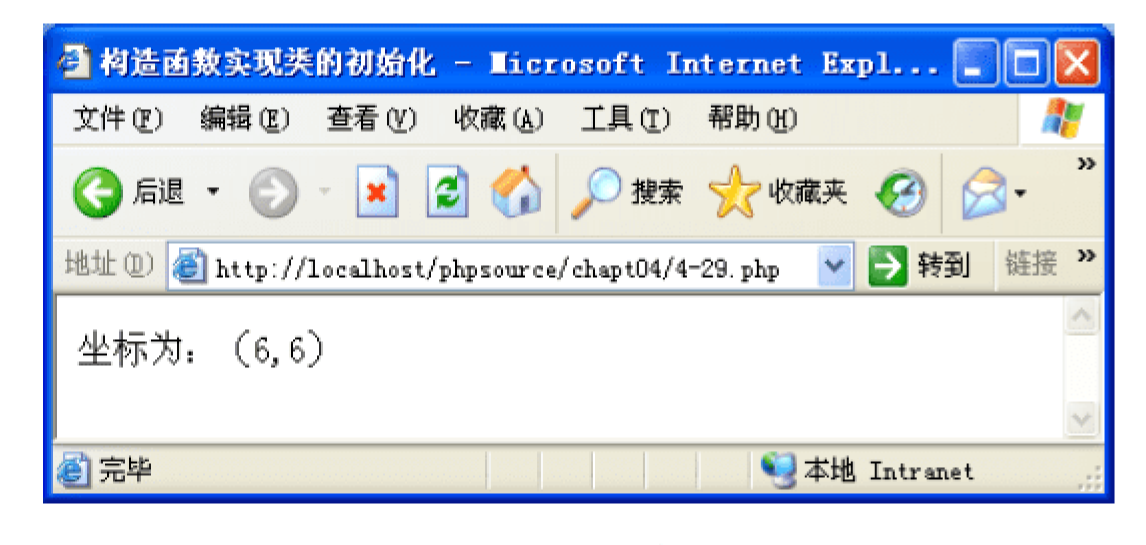


图 4-29　4-29.php 程序的运行结果

可以发现，虽然把行 31、32 的代码屏蔽了，但是点 x、y 的坐标仍然有值，这个值就是通过构造函数初始化的值。

4.6.4　类继承

PHP 5 中的类的继承功能比以前的版本有很大提高，下面就着重讲解一下使用 extends

来实现类的继承。通过对已有类的继承可以扩充类的功能，而且可以省去很多重复编码，大大降低了程序开发的复杂度。

一个类继承其他类时，可以继承其中所有的属性和方法，这种继承其他类的类称为派生类（或子类），被继承的类称为父类。创建派生类的语句如下：

```
class  派生类(子类)名  extends   父类名
```

在例 4-28 和 4-29 中已经建立了 point 类，下面就来建立它的一个派生类 circle，具体代码如下：

程序清单　文档 4-30.php

```
<!--文件 4-30.php：类的继承-->
<HTML>
 <HEAD>
      <TITLE>类的继承</TITLE>
 </HEAD>
 <BODY>
 <?php
        class point
        {
            public $x;      //点 x 的坐标
            public $y;      //点 x 的坐标
            function setx($set_x)
            {
                $this->x=$set_x;
                }
            function sety($set_y)
            {
                $this->y=$set_y;
                }
            function print_point()
            {
                echo "坐标为：(".$this->x.",".$this->y.") ";
                }
            }
        class cirle extends point{
            private $radius;
            function setradius($r){
                $this->radius=$r;
                }
            function print_radius()
             {
                echo "半径为：".$this->radius;
```

```
                }
            }
        $cirle1=new cirle;
        $cirle1->setx(6);
          $cirle1->sety(8);
          $cirle1->print_point();
          $cirle1->setradius(9);
         echo "<br>";
          $cirle1->print_radius();
?>
</BODY>
</HTML>
```

其运行结果如图 4-30 所示。

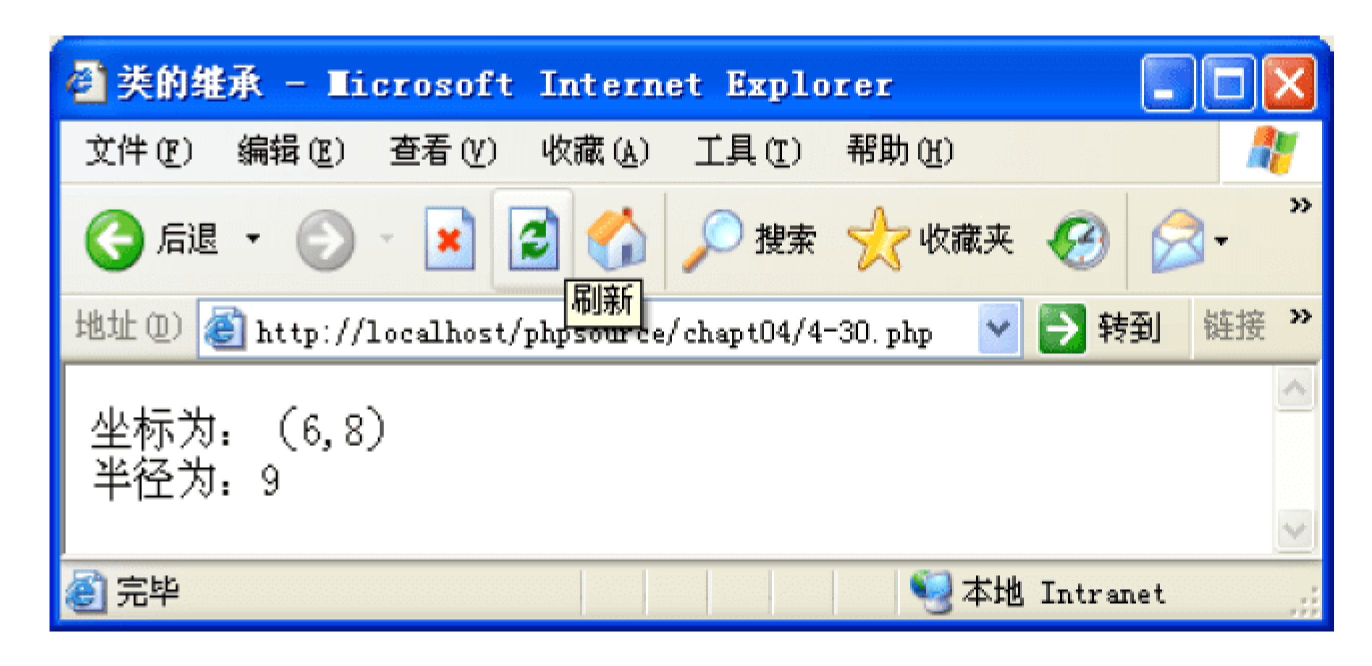


图 4-30　4-30.php 程序的运行结果

4.6.5　同名函数实现类的多态

当派生类（子类）中的某个函数的函数名与其父类中的某个函数名重复时，系统在派生类对象中会自动选用派生类中的函数，这就是类的多态。在类的继承中，对同一功能而言，子类相对于父类往往应该有一些特殊性，这些特殊性功能的实现就依赖于类的多态性。而在真正的使用中，类的多态和类的继承往往是同时出现的。具体应用举例如下：

程序清单　文档 4-31.php

```
<!--文件 4-31.php-->
<HTML>
<HEAD>
 <TITLE>类的继承与多态</TITLE>
</HEAD>
<BODY>
 <?php
     class point
     {
         public $x;    //点的 x 坐标
```

```
        public $y;    //点的 y 坐标
        function Point($x=0,$y=0)   //构造函数，初始化 x 值和 y 值
        {
            $this->x=$x;
            $this->y=$y;
        }
        function setx($px=0)   //设置 x 坐标值
        {
            $this->x=$px;
        }
        function sety($py=0)   //设置 y 坐标值
        {
            $this->y=$py;
        }
        function getx()   //得到 x 坐标值
        {
            return $this->x;
        }
        function gety()   //得到 y 坐标值
        {
            return $this->y;
        }
        function moveto($x,$y)   //移动点
        {
            if($x<0||$y<0)
            {
                echo "圆心移动失败!<br>";
                return;
            }
            $this->x=$x;
            $this->y=$y;
            echo "圆心移动成功!<br>";
        }
    }

    class circle extends point
    {
        public $radius;
        function Circle($x=10,$y=10,$r=5)   //构造函数，初始化
        {
            $this->x=$x;    //圆心 x 坐标
            $this->y=$y;    //圆心 y 坐标
            $this->radius=$r;    //圆半径
```

```
            }
            function setradius($r)
            {
                $this->radius=$r;
            }
            function getradius($r)
            {
                return $this->radius;
            }
            function moveto($x,$y)   //移动圆
            {
                if($x<($this->radius)||$y<($this->radius))
                {
                    echo "圆心移动失败!<br>";
                    return;
                }
                $this->x=$x;
                $this->y=$y;
                echo "圆心移动成功!<br>";
            }
        }
        $mypoint=new point(20,20);  //定义一个处于(20,20)的点
        $mycircle=new circle(20,20,10);
            //定义一个圆心处于(20,20),半径为 10 的圆
        $mycircle->setx(30);  //调用父类函数,改变圆心坐标 x 值
        $mypoint->moveto(5,5);  //移动点,成功
        $mycircle->moveto(5,5);  //移动圆,失败
?>
    </BODY>
</HTML>
```

其运行结果如图 4-31 所示。

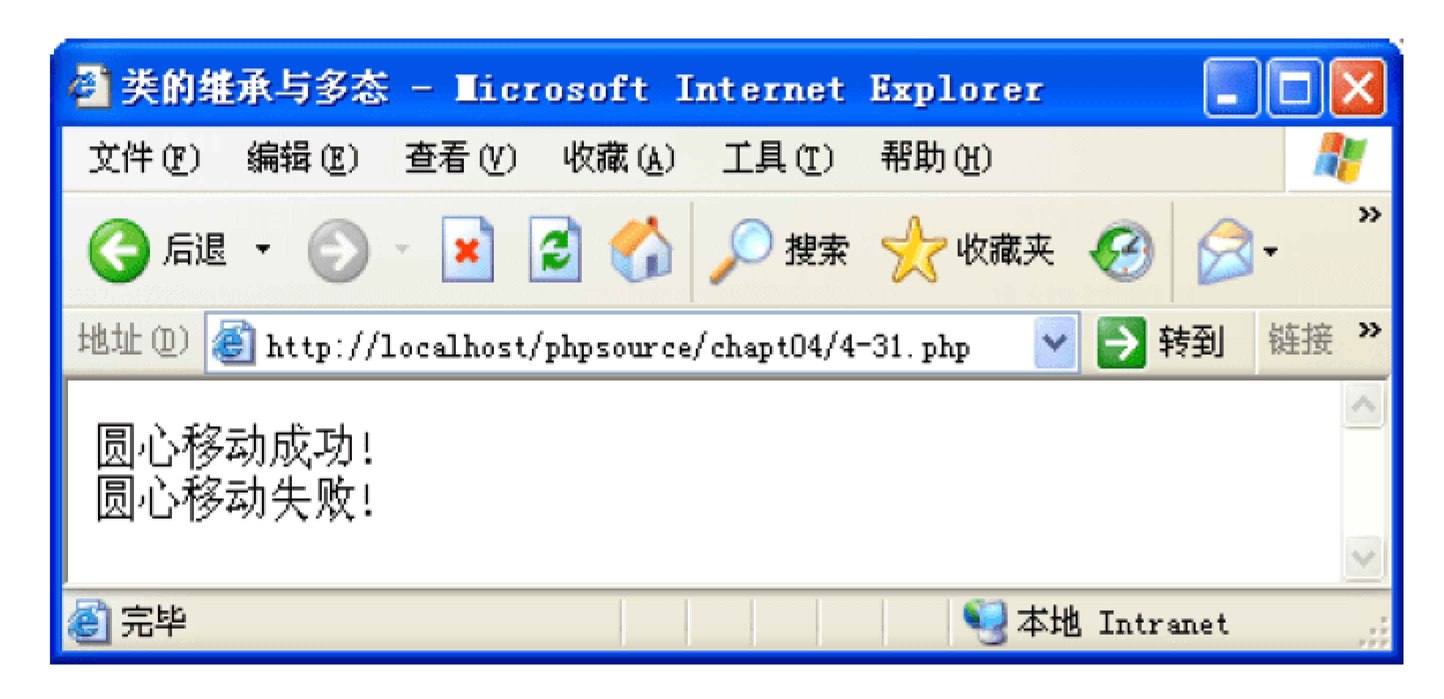


图 4-31　4-31.php 程序的运行结果

4.7 案例 4 用 PHP 编写多功能乘法表

利用类和对象的知识编写 PHP 程序，在网页上输出含九九乘法表在内的任意维的乘法表（如六乘六的乘法表）。

4.7.1 案例分析

输出九九乘法表，在任何一种语言中实现都很容易，而且它也是一个常用的例子。下面就具体分析一下如何利用类和对象的思想输出含九九乘法表在内的任意维数的乘法表。

（1）先建一个名为 table 的类，这是实现功能的主要部分；
（2）在类中定义一个公共变量$x；
（3）利用构造函数初始化乘法标的维数为 9，也就是定义公共变量$x 为 9；
（4）编写一个函数 print_table()，功能为打印出乘法表，其维数为$x，这样 table 类就编写完成；
（5）初始化对象$table1 为 table 类；
（6）对象$table1 调用其函数 print_table，这样就打印出了九九乘法表；
（7）再初始化对象$table2 也为 table 类；
（8）设置$table2 的为 6；
（9）对象$table2 调用其函数 print_table，这样就打印出了六六乘法表。

4.7.2 具体代码

通过以上分析，可以编写代码如下：

程序清单 文档 4-32.php

```
<!--文件 4-32.php：类和对象输出乘法表-->
<HTML>
 <HEAD>
     <TITLE>类和对象输出乘法表</TITLE>
 </HEAD>
 <BODY>
      <?php
          class table
          {
              public $x;      //乘法表的维数
               function table()
               {        //构造函数初始化乘法表的维数
                   $this->x=9;
                   }
               function print_table()
               {
```

```
                  for($i=1;$i<=$this->x;$i++){
                      for($j=1;$j<=$i;$j++){
                          echo $j."*".$i." ";
                         }
                      echo "<br>";
                     }
                  }    //end function print_table()
              } //end class
          $table1=new table;
          $table1->print_table();
          echo "<br>";
          $table2=new table;
          $table2->x=6;
         $table2->print_table();
     ?>
</BODY>
</HTML>
```

其运行效果如图 4-32 所示。

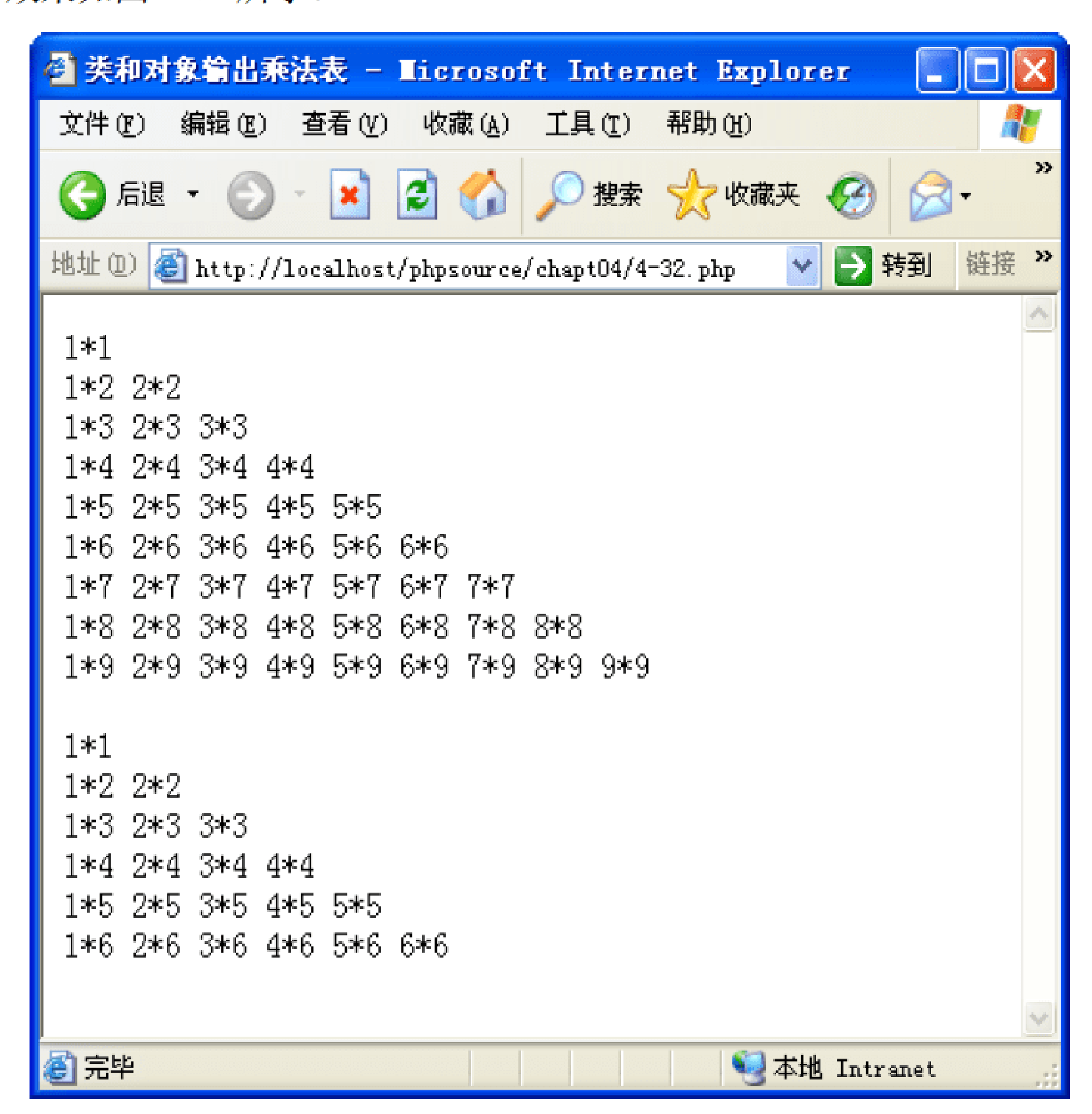


图 4-32　4-32.php 程序的运行结果

4.8 本章小结

本章着重介绍了 PHP 5 的程序结构、变量和常量、运算符和表达式，PHP 5 的程序流程控制和自定义函数以及类和对象的基本概念和应用。这些知识都是学好 PHP 5 的基础。学习过 C 语言的读者应该发现，本章内容虽多，但难度不大。值得注意的是如果将 PHP 5 中的自定义函数以及类和对象掌握好了，并能运用到实际开发中，特别是类和对象思想运用熟练了，将会使繁琐的开发过程和模块化进程大大简化，达到事半功倍的效果。

4.9 练习题

1. PHP 5 支持哪些数据类型？
2. PHP 5 中的操作符有哪些类型？
3. PHP 5 中有哪几种流程结构？
4. PHP 5 中实现分支结构和循环结构都有哪些语句？
5. PHP 5 中如何定义类和对象？
6. PHP 5 的类中变量有几种类型？
7. PHP 5 中如何使用对象？
8. 仿照文件 4-32.php，把文件 4-16.php 改写成用类和对象的方法实现相同的功能的程序。

CHAPTER 5

PHP 5 内置数组和常量

本章重点

- 接收表单数据和 URL 附加数据
- 用 Cookie 和 Session 实现用户登录
- 用内置数组获取服务器环境信息

PHP + MySQL

5.1 PHP 5 的内置数组简介

PHP 提供了一套附加的内置数组（也称为预定数组），这些数组变量包含了来自 Web 服务器（如果可用）、运行环境和用户输入的数据。这些数组非常特别，它们在全局范围内自动生效，因此通常被称为自动全局变量（autoglobals）或者超全局变量（superglobals）（PHP 中没有用户自定义超全局变量的机制）。超全局变量和旧的预定义数组（$HTTP_*_VARS）同时存在。而且，自 PHP 5.0.0 起，长格式的 PHP 预定义变量可以通过设置 register_long_arrays 来屏蔽。下面简单介绍最常用的 10 个内置数组。

1. $GLOBALS（Global 变量）

包含一个引用指向每个当前脚本的全局范围内有效的变量。该数组的键名为全局变量的名称。

2. $_SERVER（服务器变量）

变量由 Web 服务器设定或者直接与当前脚本的执行环境相关联。类似于旧数组 $HTTP_SERVER_VARS 数组（依然有效，但反对使用）。

3. $_GET（HTTP GET 变量）

经由 URL 请求提交至脚本的变量。类似于旧数组 $HTTP_GET_VARS 数组（依然有效，但反对使用）。

4. $_POST（HTTP POST 变量）

经由 HTTP POST 方法提交至脚本的变量。类似于旧数组 $HTTP_POST_VARS 数组（依然有效，但反对使用）。

5. $_COOKIE（HTTP Cookies）

经由 HTTP Cookies 方法提交至脚本的变量。类似于旧数组 $HTTP_COOKIE_VARS 数组（依然有效，但反对使用）。

6. $_FILES（HTTP 文件上传变量）

经由 HTTP POST 文件上传而提交至脚本的变量。类似于旧数组 $HTTP_POST_FILES 数组（依然有效，但反对使用）。

7. $_ENV（环境变量）

执行环境提交至脚本的变量。类似于旧数组 $HTTP_ENV_VARS 数组（依然有效，但反对使用）。

8. $_REQUEST（Request 变量）

经由 GET，POST 和 COOKIE 机制提交至脚本的变量，该数组并不值得信任（建议尽量少用，甚至不用）。所有包含在该数组中的变量的存在与否以及变量的顺序均按照

php.ini 中的 variables_order 配置指示来定义。

9. $_SESSION（Session 变量）

当前注册给脚本会话的变量。类似于旧数组$HTTP_SESSION_VARS 数组（依然有效，但反对使用）。

10. $php_errormsg（前一个错误消息）

包含 PHP 产生的上一错误消息内容的变量。该变量在发生错误并且 track_errors 选项打开（默认为关闭）后才有效。

下面的章节就常用的内置数组做较详细的讲解。

5.2　接收表单数据和 URL 附加数据

网页的数据传递不外乎有两种方式，一种是接收表单数据（也称为 POST 方法），另一种是 URL 附加数据（也称为 GET 方法）。下面先来学习一下表单接收数据的方法，请看下例。

程序清单　文档 5-1.php

```
<!--文件 5-1.php：表单数据传递-->
<HTML>
 <HEAD>
     <TITLE>表单数据传递</TITLE>
 </HEAD>
 <BODY>
   <?php
   $tag=$_POST["tag"];
   if($tag==1){
       $addend1=$_POST["addend1"];
        $addend2=$_POST["addend2"];
        }else{
            $addend1=0;
            $addend2=0;
            }
    $sum=$addend1+$addend2;
    ?>
 请在下面的表单中输入两数以求其和
 <form name="form1" method="post" action="#">
     <input type="hidden" name="tag" size="4" value="1">
  <input type="text" name="addend1" size="4" value="<?php echo $addend1;?>">
  +
  <input type="text" name="addend2" size="4" value="<?php echo
```

```
        $addend2;?>">
24: =
25:       <?php echo $sum;?><br>
26: <br><input type="submit" name="Submit" value="计算">
27: <input type="reset" name="Submit2" value="重置">
28: </form>
29: </BODY>
30:</HTML>
```

对于上面的程序，不难发现从第 19 行到 28 行都不是 PHP 的程序，而是 HTML 中的表单代码，其中第 20 行是一个初始值为 1 的隐藏表单，它在浏览器中是不显示的，但是也可以随着单击“计算”按钮的提交而传递数据。注意，隐藏表单在网页中的用途是非常重要而且十分广泛的。第 7 行到第 17 行是程序的主要部分，先用 POST 方法接收隐藏表单的数据，然后判断如果为 1，也就是单击过“计算”按钮之后的页面，就接收两个加数，否则，就初始化两个加数为 0，第 16 行计算两个加数的和。

当没有输入数据时，也就是初次浏览时，其运行结果如图 5-1（a）所示。



图 5-1 （a） 5-1.php 程序的运行结果

当在两个表单里输入“88”和“66”两个数值并单击“计算”按钮后，其运行结果如图 5-1（b）所示。



图 5-1 （b） 5-1.php 程序的运行结果

接下来，再看一下 URL 附加数据时怎么接收的。请看下面的例子：

程序清单　文档 5-2.php

```
1: <!--文件 5-2.php：URL 附加数据传递-->
2: <HTML>
3:  <HEAD>
4:      <TITLE>URL 附加数据传递</TITLE>
5:  </HEAD>
6:  <BODY>
7:  <a href="5-2.php?show_tag=1">显示图片</a>
8:  <a href="?show_tag=2">隐藏图片</a><br><br>
9:  <?php
10: $show_tag=$_GET["show_tag"];
11: if($show_tag==1){
12: echo "<img src=ditu.jpg width=128 height=96 align=left>";
13: }
14: ?>
15:
16: </BODY>
17: </HTML>
```

分析上面的例子，可以发现：第 7 行和第 8 行均为 HTML 中的超链接，两者不同的是，第 7 行为链接到文件“5-2.php”并附加数据“show_tag=1”，第 8 行为链接到本页文件并附加数据“show_tag=2”，但这两行的得最终效果是一样的。第 10 行是接收 URL 附件数据的具体方法。第 11 行到第 13 行是如果接收过来的 URL 的附件数据为 1 时输出显示图片的 HTML 代码。

当单击“显示图片”时，其运行效果如图 5-2（a）所示。

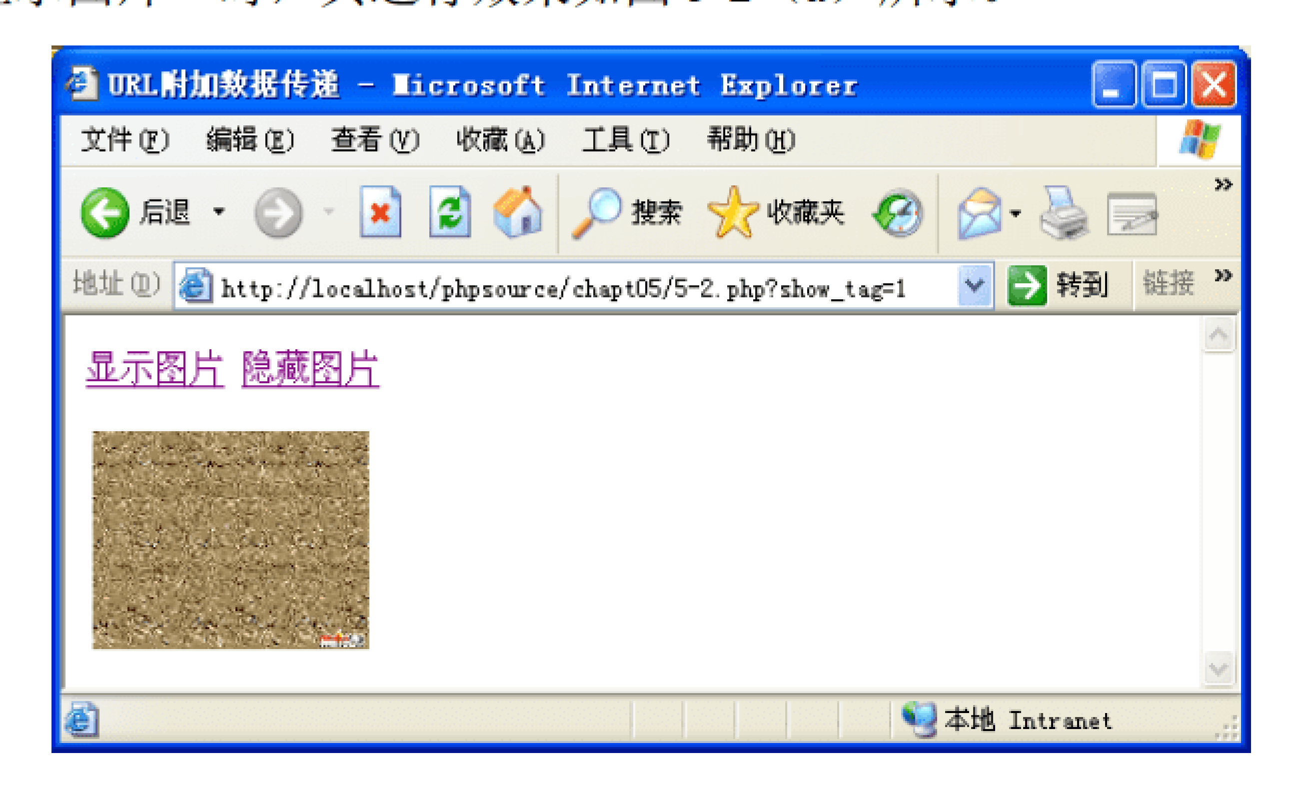


图 5-2（a）　5-2.php 程序的运行结果

当单击“隐藏图片”时，其运行效果如图 5-2（b）所示。

上面的例子，仅仅传递了一个数据，如果要传递多个数据应该怎么做呢？只需要把第 7 行的“5-2.php?show_tag=1”代码改写成诸如“5-2.php?show_tag=1&date_name=date”的代码，就传递了两个数据，也就是说在传递多个数据的时候要用“&”进行连接。

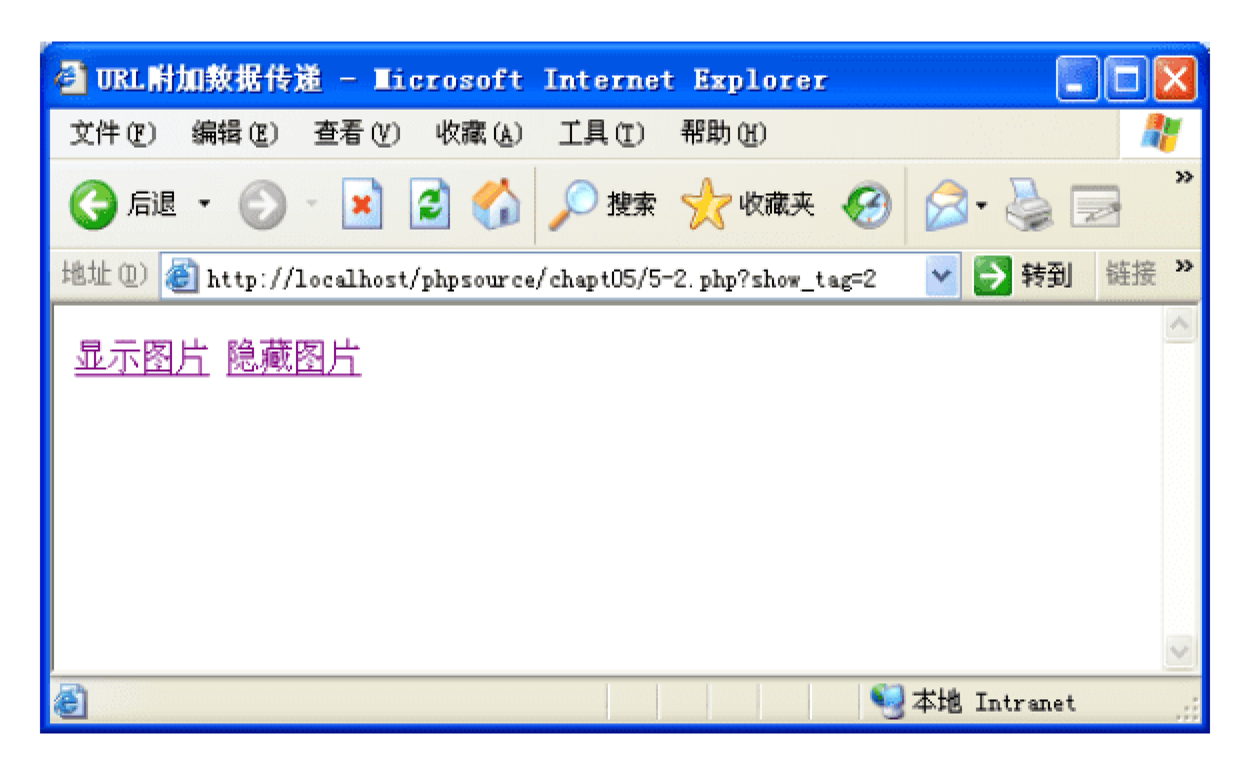


图 5-2 （b） 5-2.php 程序的运行结果

通过上面的两个例子，不难发现两种数据传递方法的不同点：GET 方法可以在地址栏中显示出来，比较暴露，存在着安全隐患；而 POST 方法则比较隐蔽，特别是隐藏表单的使用，不仅可以增加程序的安全性，还可以传递一些不需要用户输入或不能让用户更改的贯穿若干个网页之间的量值。

提 示

> 有了这两种数据传递的方法，是不是就可以完成许许多多交互功能了呢？GET 方法的数据，是否可以通过其他的方法来增加程序的安全性呢？如果传递诸如“id=34&del_id=56”，肯定会被人知道传递的是数据表或其他的 id 和 del_id 而且其值为 34 和 56，但是如果传递“34,56”或“34-56”，谁又会知道我们传递的是什么东西呢？而在接收到数据的时候，只要根据需要进行字符截取就可以使用了。这样不就大大提高了 GET 方法数据传递的安全性了吗？

5.3 用 Cookie 和 Session 实现用户登录

Cookie 和 Session 在 Web 技术中占有非常重要的份量。由于网页是一种无状态的连接程序，你无法得知用户的浏览状态，因此必须通过 Cookie 或 Session 记录用户的有关信息，以便用户再次以此身份对 Web 服务器提出要求时作确认。例如，在某些网站中常常要求用户登录，但怎么知道用户已经登录了呢，如果没有 Cookie 和 Session 的话，登录信息是无法保留的，那岂不要让用户在每一页网页中都提供用户名和密码。

Cookie 是一种在远程浏览器端储存数据并以此来跟踪和识别用户的机制。PHP 透明地支持 HTTP Cookie。从客户端发送的 Cookie 都会被 PHP 5 自动包括进$_COOKIE 的全局数组。如果希望对一个 Cookie 变量设置多个值，则需在 Cookie 的名称后加“["值名称"]”符号。

Session 储存于服务器端（默认以文件方式存储 session），根据客户端提供的 session id 来得到用户的文件，取得变量的值，session id 可以通过客户端的 Cookie 或者访问的 URL 的附加数据来传送给服务器，然后服务器读 Session 的目录。也就是说，session id 是取得存储在服务上 Session 变量的身份证。如果配置文件 php.ini 中没有设置“session.auto_start=1”，

那么要使用 Session 就必须先运行代码 session_start()。运行的时候，在服务器上产生了一个 Session 文件，随之也产生了与之惟一对应的一个 session id，定义 session 变量以一定形式存储在刚才产生的 Session 文件中。通过 session id，可以取出定义的变量。

PHP 中的 Session 在默认情况下是使用客户端的 Cookie 来保存 session id 的,所以当客户端的 Cookie 出现问题的时候就会影响 Session 了。必须注意的是：Session 不一定必须依赖 Cookie，这也是 Session 相比 Cookie 的高明之处。当客户端的 Cookie 被禁用或出现问题时，PHP 会自动把 session id 附着在 URL 附加数据中,这样再通过 session id 就能跨页使用 Session 变量了。

下面先来看一下 Cookie 的具体应用实例。

先建立一个用户登录的表单：

程序清单　文档 5-3.php

```
<!--文件 5-3.php：COOKIE 实现用户登录的表单-->
<HTML>
 <HEAD>
     <TITLE>COOKIE 实现用户登录的表单</TITLE>
 </HEAD>
 <BODY>
 <form name="form1" method="post" action="5-3action.php">
 <table width="280" height="96" border="0" align="center"
cellpadding="0" cells: pacing="1" bgcolor="#999999">
    <tr>
          <td colspan="2" align="center" bgcolor="#FFFFFF">用户登录</td>
         </tr>
         <tr>
           <td align="right" bgcolor="#FFFFFF">用户名:</td>
           <td align="left" bgcolor="#FFFFFF">
              <input type="text" name="user_name" size="12">
           </td>
         </tr>
         <tr>
          <td align="right" bgcolor="#FFFFFF">口令:</td>
          <td align="left" bgcolor="#FFFFFF">
           <input type=" password " name="user_pw" size="12"></td>
         </tr>
         <tr>
           <td colspan="2" align="center" bgcolor="#FFFFFF">
           <input type="submit" name="Submit" value="提交"> 
           <input type="reset" name="Submit2" value="重置"></td>
         </tr>
        </table>
 </form>
```

```
30:     </BODY>
31:  </HTML>
```

运行效果如图 5-3（a）所示。

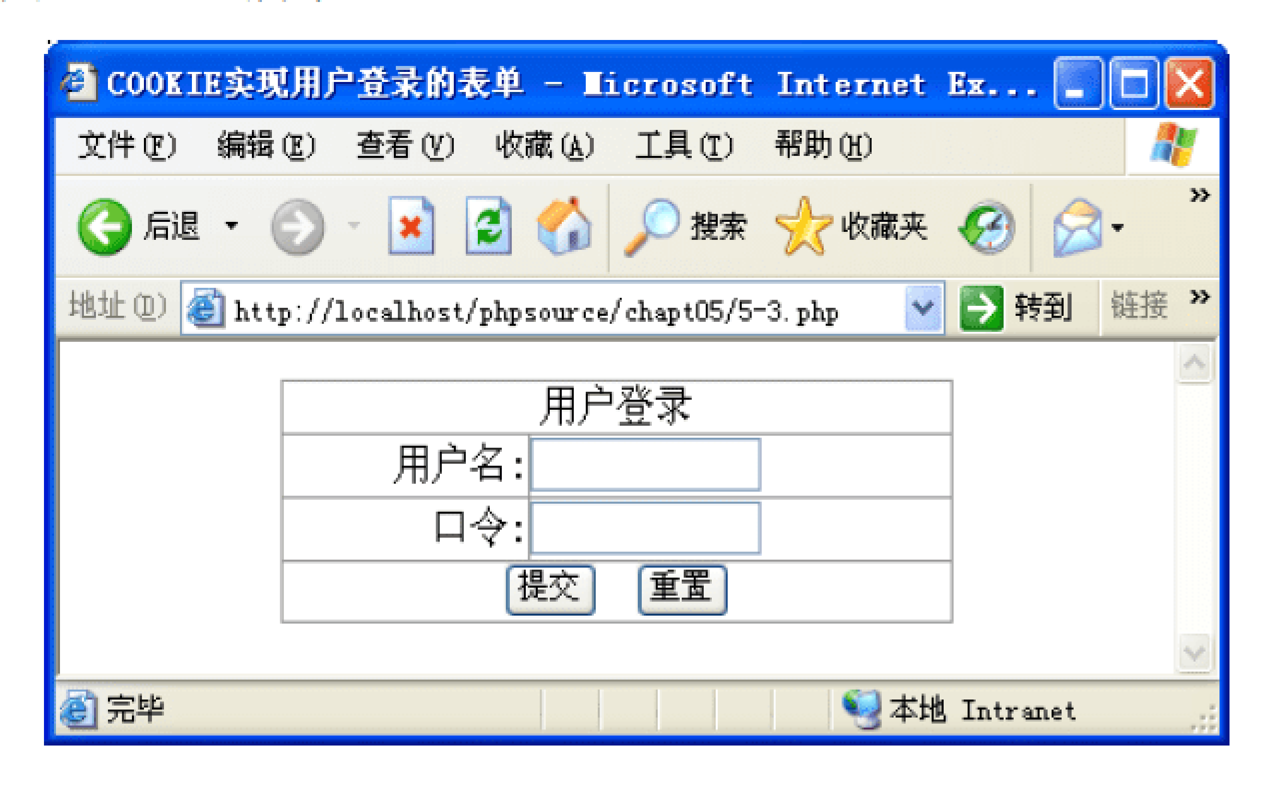


图 5-3（a） 5-3.php 程序的运行结果

下面再建立一个名叫“5-3action.php”的文件，在“5-3.php”文件中的第 7 行已经规定图 5-3（a）中用户填写的表单内容要提交到“5-3action.php”文件中。也就是在文件“5-3action.php”中，我们利用 Cookie 模拟用户的登录是否成功。详细代码如下：

程序清单 文档 5-3action.php

```
1: <!--文件 5-3action.php：COOKIE 实现用户登录-->
2: <HTML>
3:      <HEAD>
4:          <TITLE>COOKIE 实现用户登录</TITLE>
5:      </HEAD>
6:      <BODY>
7:      <?php
8:      $_COOKIE["user_name"]=$_POST["user_name"];
9:      $_COOKIE["user_pw"]=$_POST["user_pw"];
10:     if($_COOKIE["user_name"]=="php" && $_COOKIE["user_pw"]=="php5"){
11:     echo "恭喜您，用户名和口令正确，登录成功！";
12: }else{
13:         echo "您输入的用户名为：".$_COOKIE["user_name"];
14:         echo "<br>口令为：".$_COOKIE["user_pw"];
15:         echo "<br>但是不正确，请尝试：php 和 php5";
16:     }
17:     ?>
18:     <br><a href="5-3action-check.php">单击检测 Cookie 的值是否可以页间传递</a>
19:       </BODY>
19: </HTML>
```

第 8 行和第 9 行接收“5-3.php”程序中第 15 行和第 21 行表单传递的变量值，并注册为

Cookie，此时“$_COOKIE["user_name"]”和“$_COOKIE["user_pw"]”均为全局变量，只要是由程序“5-3action.php”生成的和跳转过去的页面，这两个变量的值均可用。

当在图 5-3（a）中输入用户名 php 和密码“php5”后，单击“提交”按钮，会出现如图 5-3（b）所示的运行效果。

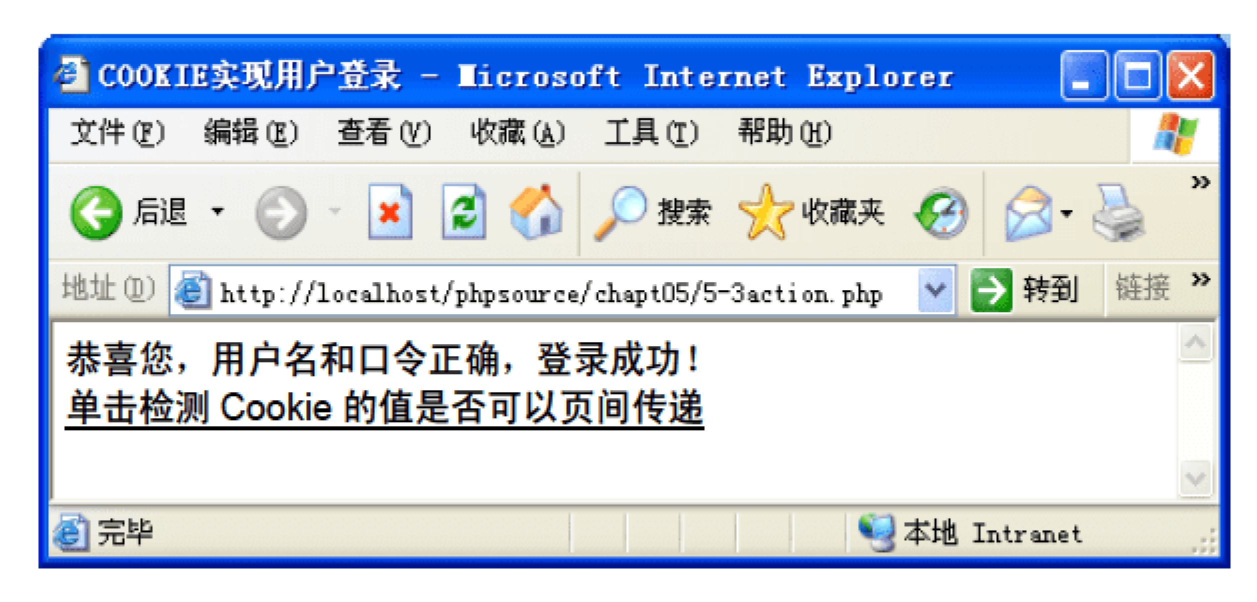


图 5-3（b）　5-3action.php 程序的运行结果

当输入其他用户名和密码后，单击“提交”按钮，会出现如图 5-3（c）所示的运行效果。

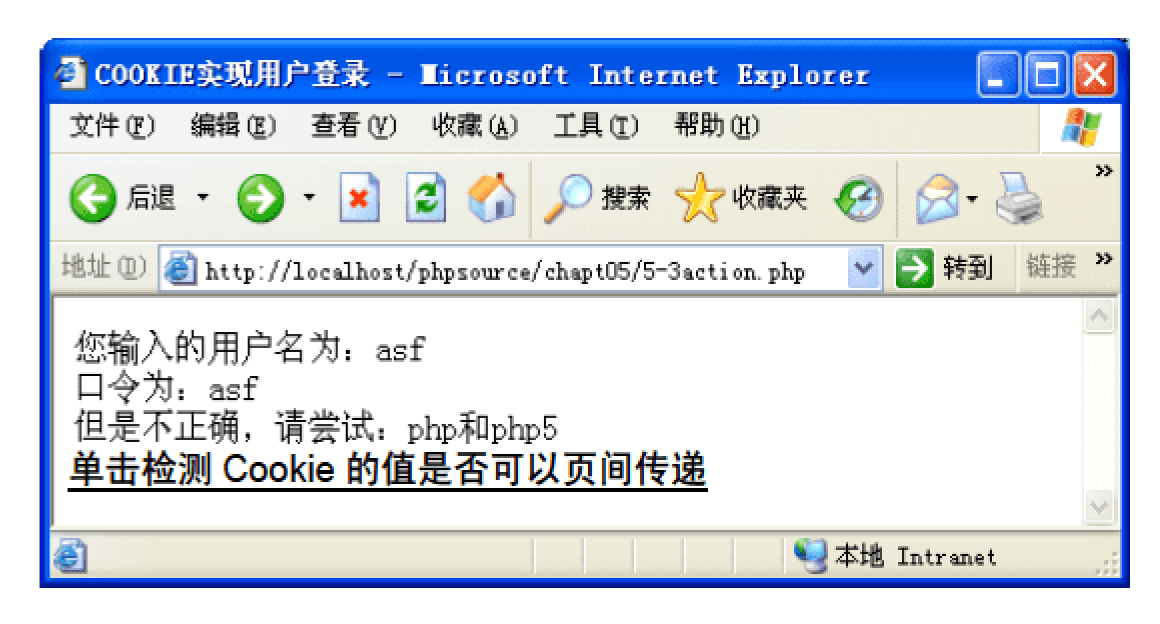


图 5-3（c）　5-3action.php 程序的运行结果

下面再来检测一下 Cookie 在系统设为禁用的时候是否可以在不同页间传递。建立一个名为“5-3action-check.php”的文件，在文件“5-3action.php”的第 18 行，我们需单击一个链接方可进行测试，而链接的目标文件为“5-3action.php-check.php”，文件“5-3action.php-check.php”的详细代码如下：

程序清单　文档 5-3action-check.php

```
<!--文件 5-3action-check.php：COOKIE 页间传递-->
<HTML>
 <HEAD>
     <TITLE>COOKIE 页间传递</TITLE>
 </HEAD>
 <BODY>
 <?php
   if($_COOKIE["user_name"]!="" && $_COOKIE["user_pw"]!=""){
       echo "Cookie 页间传递成功！<br>";
       echo "您输入的用户名为：".$_COOKIE["user_name"];
```

```
        echo "<br>口令为：".$_COOKIE["user_pw"];
    }else{
        echo "Cookie 页间传递失败，其值为空！<br>";
        }
?>
</BODY>
</HTML>
```

打开 IE“工具”菜单上的“Internet 选项”→“安全”→“自定义级别”，将安全设置中的“允许使用每个对话 cookies”设为禁用，则“5-3action.php-check.php”运行效果如图 5-3（d）所示。

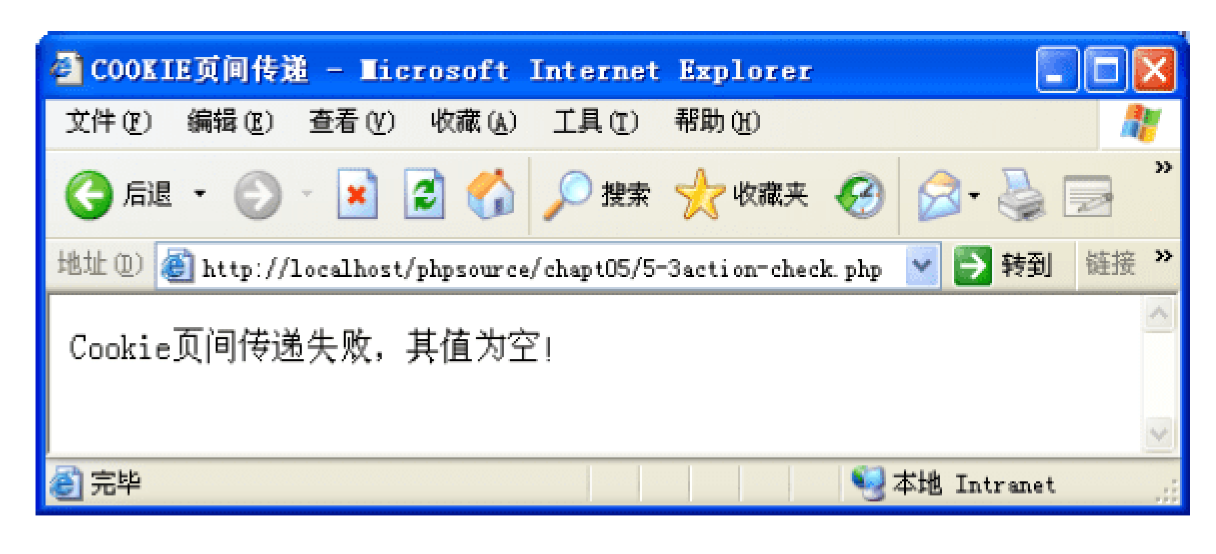


图 5-3（d） 5-3action-check.php 程序的运行结果

把文件“5-3.php”、“5-3action.php”、“5-3action-check.php”中所有的“$_COOKIE”替换为“$_SESSION”，“Cookie”替换为“Session”，以及链接地址更改后（详见本书配套源码），分别存储为新文件“5-4.php”、“5-4action.php”、“5-4action-check.php”。“5-4action-check.php”的运行效果如图 5-4 所示。

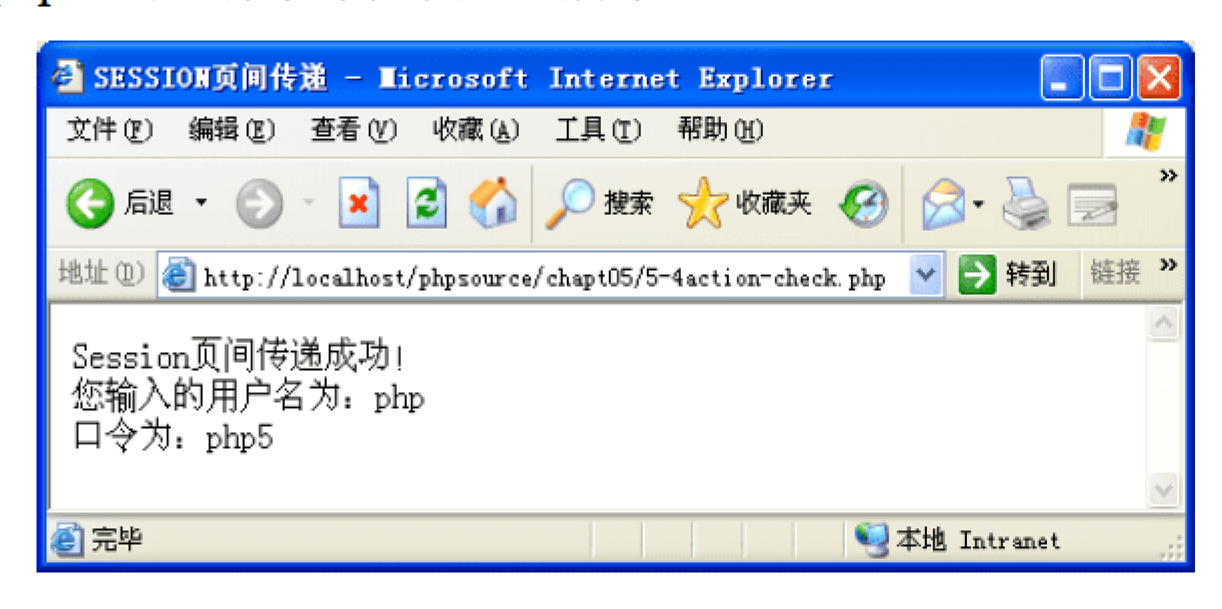


图 5-4 5-4action-check.php 程序的运行结果

5.4 用内置数组获取服务器环境信息

$_SERVER 是一个包含诸如头信息（header）、路径（path）和脚本位置（script locations）的数组。数组的实体由 Web 服务器创建，但是并不能保证所有的服务器都能产生全部的信息，服务器可能忽略了一些信息，或者产生了一些未在下面列出的新的信息。

$_SERVER 是一个 superglobal 变量，也就是自动全局变量。这就意味它在所有的脚本中都有效，而且在函数或方法中不需要使用 global $_SERVER，直接访问就可以了。

要想访问$_SERVER 中的某一个信息，可以用下面的方式：

```
$_SERVER["信息名称"]或$_SERVER['信息名称']
```

$_SERVER 中具体的信息名称如表 5-1 所示。

表 5-1　$_SERVER 中的信息

信息名称	信息含义
PHP_SELF	当前正在执行脚本的文件名，与 document root 相关。举例来说，在 URL 地址为 http://localhost/phpsource/chapt05/5-5.php 的脚本中使用 $_SERVER['PHP_SELF'] 将会得到 /phpsource/chapt05/5-5.php 这个结果。
argv	传递给该脚本的参数。当脚本运行在命令行方式时，argv 变量传递给程序 C 语言样式的命令行参数。当调用 GET 方法时，该变量包含请求的数据。
argc	包含传递给程序的命令行参数的个数（如果运行在命令行模式）。
GATEWAY_INTERFACE	服务器使用的 CGI 规范的版本。例如，“CGI/1.1”。
SERVER_NAME	当前运行脚本所在服务器主机的名称。如果该脚本运行在一个虚拟主机上，该名称是由那个虚拟主机所设置的值决定。
SERVER_SOFTWARE	服务器标识的字串，在响应请求时的头信息中给出。
SERVER_PROTOCOL	请求页面时通信协议的名称和版本。例如，“HTTP/1.0”。
REQUEST_METHOD	访问页面时的请求方法。例如：“GET”、“HEAD”，“POST”，“PUT”。
REQUEST_TIME	请求开始时的时间戳。从 PHP5.1.0 起有效。
QUERY_STRING	查询（query）的字符串（URL 中第一个问号 ? 之后的内容）。
DOCUMENT_ROOT	当前运行脚本所在的文档根目录。在服务器配置文件中定义。
HTTP_ACCEPT	当前请求的 Accept:头信息的内容。
HTTP_ACCEPT_CHARSET	当前请求的 Accept-Charset:头信息的内容。例如：“iso-8859-1,*,utf-8”。
HTTP_ACCEPT_ENCODING	当前请求的 Accept-Encoding:头信息的内容。例如：“gzip”。
HTTP_ACCEPT_LANGUAGE	当前请求的 Accept-Language:头信息的内容。例如：“en”。
HTTP_CONNECTION	当前请求的 Connection:头信息的内容。例如：“Keep-Alive”。
HTTP_HOST	当前请求的 Host:头信息的内容。
HTTP_REFERER	链接到当前页面的前一页面的 URL 地址。不是所有的用户代理（浏览器）都会设置这个变量，而且有的还可以手工修改 HTTP_REFERER。因此，这个变量不总是真实正确的。
HTTP_USER_AGENT	当前请求的 User-Agent:头信息的内容。该字符串表明了访问该页面的用户代理的信息。一个典型的例子是：Mozilla/4.5 [en](X11; U; Linux 2.2.9 i586)。也可以使用 get_browser()得到此信息。
HTTPS	如果脚本是通过 HTTPS 协议被访问，则被设为一个非空的值。
REMOTE_ADDR	正在浏览当前页面用户的 IP 地址。这个是经常用在记录访问用户的 IP 中。

（续表）

信息名称	信息含义
REMOTE_HOST	正在浏览当前页面用户的主机名。反向域名解析基于该用户的 REMOTE_ADDR。
REMOTE_PORT	用户连接到服务器时所使用的端口。
SCRIPT_FILENAME	当前执行脚本的绝对路径名。
SERVER_ADMIN	该值指明了 APACHE 服务器配置文件中的 SERVER_ADMIN 参数。如果脚本运行在一个虚拟主机上，则该值是那个虚拟主机的值。
SERVER_PORT	服务器所使用的端口。默认为“80”。如果使用 SSL 安全连接，则这个值为用户设置的 HTTP 端口。
SERVER_SIGNATURE	包含服务器版本和虚拟主机名的字符串。
PATH_TRANSLATED	当前脚本所在文件系统（不是文档根目录）的基本路径。这是在服务器进行虚拟到真实路径的映像后的结果。
SCRIPT_NAME	包含当前脚本的路径。这在页面需要指向自己时非常有用。__FILE__ 包含当前文件的绝对路径和文件名（例如包含文件）。
REQUEST_URI	访问此页面所需的 URI。例如，“/index.html”。
PHP_AUTH_DIGEST	当作为 APACHE 模块运行时，进行 HTTP Digest 认证的过程中，此变量被设置成客户端发送的“Authorization”HTTP 头内容（以便作进一步的认证操作）。
PHP_AUTH_USER	当 PHP 运行在 APACHE 或 IIS（PHP 5 是 ISAPI）模块方式下，并且正在使用 HTTP 认证功能，这个变量便是用户输入的用户名。
PHP_AUTH_PW	当 PHP 运行在 APACHE 或 IIS（PHP 5 是 ISAPI）模块方式下，并且正在使用 HTTP 认证功能，这个变量便是用户输入的密码。
AUTH_TYPE	当 PHP 运行在 APACHE 模块方式下，并且正在使用 HTTP 认证功能，这个变量便是认证的类型。

下面是一个具体的应用实例：

程序清单　文档 5-5.php

```
<!--文件 5-5.php：服务器信息的获取-->
<HTML>
     <HEAD>
         <TITLE>服务器信息的获取</TITLE>
     </HEAD>
     <BODY>
     <?php
      echo "1、".$_SERVER["PHP_SELF"]."<br>";
      echo "2、".$_SERVER["argv"]."<br>";
      echo "3、".$_SERVER["argc"]."<br>";
      echo "4、".$_SERVER["SERVER_NAME"]."<br>";
      echo "5、".$_SERVER["SERVER_SOFTWARE"]."<br>";
```

```
        echo "6、".$_SERVER["SERVER_PROTOCOL"]."<br>";
        echo "7、".$_SERVER["REQUEST_METHOD"]."<br>";
        echo "8、".$_SERVER["REQUEST_TIME"]."<br>";
        echo "9、".$_SERVER["QUERY_STRING"]."<br>";
        echo "10、".$_SERVER["DOCUMENT_ROOT"]."<br>";
        echo "11、".$_SERVER["HTTP_ACCEPT"]."<br>";
        echo "12、".$_SERVER["HTTP_ACCEPT_ENCODING"]."<br>";
        echo "13、".$_SERVER["HTTP_ACCEPT_LANGUAGE"]."<br>";
        echo "14、".$_SERVER["HTTP_CONNECTION"]."<br>";
        echo "15、".$_SERVER["HTTP_HOST"]."<br>";
        echo "16、".$_SERVER["HTTP_USER_AGENT"]."<br>";
        echo "17、".$_SERVER["HTTPS"]."<br>";
        echo "18、".$_SERVER["REMOTE_ADDR"]."<br>";
        echo "19、".$_SERVER["REMOTE_HOST"]."<br>";
        echo "20、".$_SERVER["SCRIPT_FILENAME"]."<br>";
        echo "21、".$_SERVER["SERVER_PORT"]."<br>";
        echo "22、".$_SERVER["PATH_TRANSLATED"]."<br>";
        echo "23、".$_SERVER["SCRIPT_NAME"]."<br>";
        echo "24、".$_SERVER["REQUEST_URI"]."<br>";
      ?>
     </BODY>
 </HTML>
```

在浏览器的地址栏中输入“http://localhost/phpsource/chapt05/5-5.php”时其运行效果如图 5-5（a）所示，在浏览器的地址栏中输入“http://localhost/phpsource/chapt05/5-5.php?id=8”时其运行效果如图 5-5（b）所示。同样在浏览器的地址栏中分别输入“http://127.0.0.1/phpsource/chapt05/5-5.php”和“http:// 127.0.0.1/phpsource/chapt05/5-5.php?id=8”时，又会得到不完全相同的运行效果。对此图 5-5（a）和图 5-5（b）可以看出，不同服务器的信息在有的时候输出效果是一样的，如图 5-5（a）中的 23 行和 24 行。有的服务器信息在某些时候是没有输出值的，而当地址栏信息发生变换时就有了输出值，如图 5-5（a）和图 5-5（b）中的第 9 行。

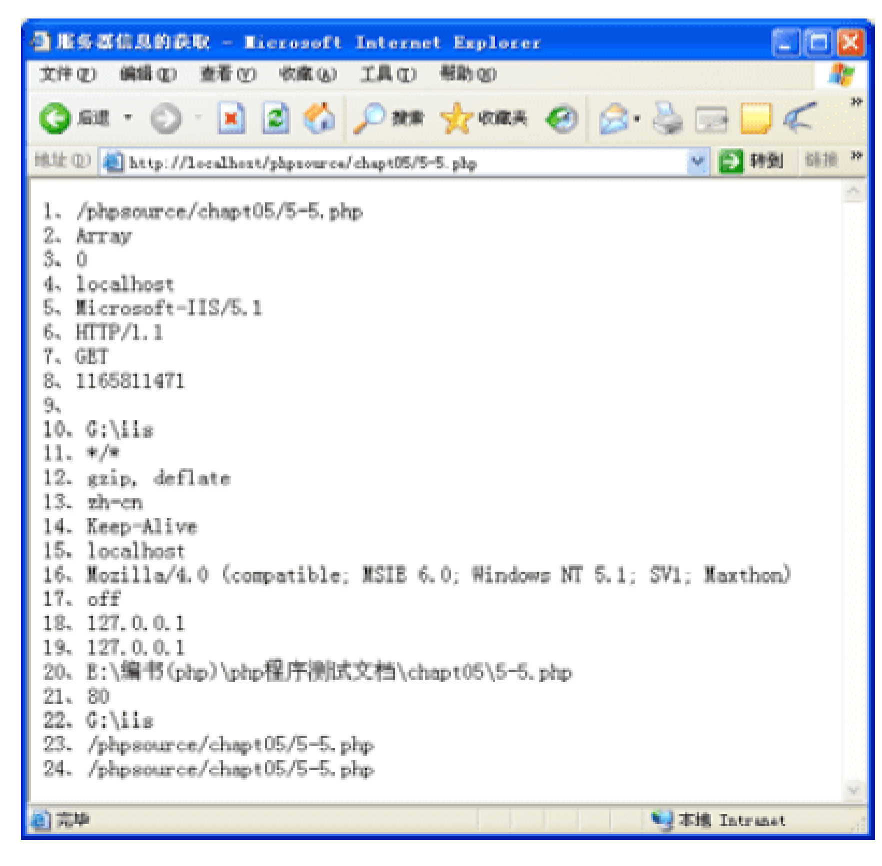


图 5-5（a）　5.5.php 程序的运行结果

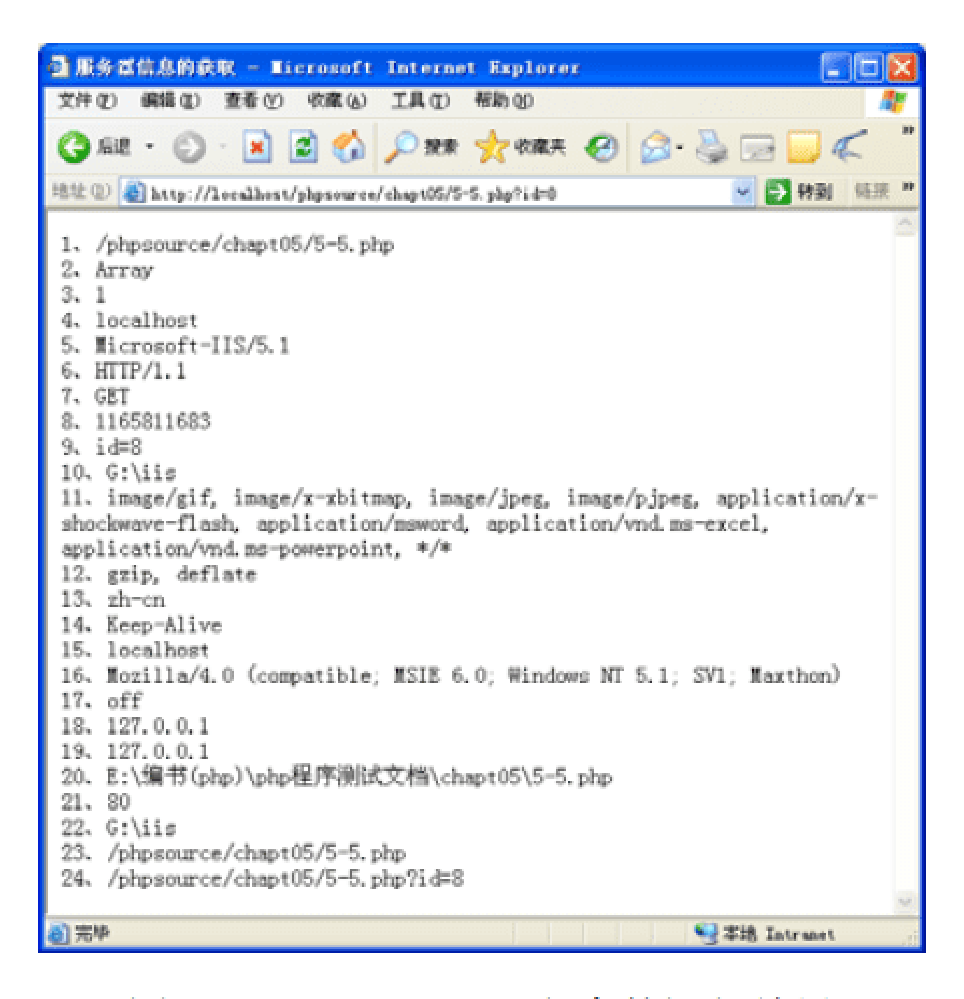


图 5-5（b） 5.5.php 程序的运行结果

5.5 案例 5 用 PHP 内置数组接收表单数据

在案例 2 的基础上，编写 PHP 程序接收用户输入的注册信息，用 PHP 作输入合法性验证，并输出用户的注册信息。

5.5.1 案例分析

仔细分析本案例，实际上就是变量接收和条件语句，具体制作步骤如下：

（1）设计一简单的用户自我介绍界面，即设计提交表单，在第 2 章的案例中已经给出；

（2）接收表单中所有的变量；

（3）用条件语句判断变量是否合法，如果合法就输出，否则给出错误信息并给出返回按钮。

5.5.2 具体代码

从分析可以得出需设计两个页面，也就是两个 php 文件。第一个页面在第 2 章中已经给出，即 anli1.php，第二个页面也就是表单接收判断页面，命名为 anli2.php。

anli1.php 代码为第 2 章中去掉第 4-26 行的代码，就不再列出。anli2.php 的代码如下：

程序清单 文档 anli2.php

```
<html>
<head>
<title>注册验证</title>
</head>
<body>
<div align="center">
  <?php
```

```
//接收 anli1.php 中的数据
$name=$_POST["name"];
$sex=$_POST["sex"];
$age=$_POST["age"];
$password=$_POST["password"];
$password1=$_POST["password1"];
$like1=$_POST["like1"];
$like2=$_POST["like2"];
$like3=$_POST["like3"];
$like4=$_POST["like4"];
$color=$_POST["color"];
$jieshao=$_POST["jieshao"];
//验证表单数据的合法性
if($name==""){
    echo "===名字不能为空，请<a href=javascript:history.go(-1)>返回</a>重新输入!===";
    exit;
}
    else if($age<1 or $age>100){
        echo "===年龄不再 0 到 100 之间，请<a href=javascript:history.go(-1)>返回</a>重新输入!===";
        exit;
    }
    else if(strlen($password)<4 or strlen($password)>10 or strlen($password1)<4 or strlen($password1)>10 or $password!=$password1){
        echo "===两次密码长度均在 4-10 之间且要相等，请<a href=javascript:history.go(-1)>返回</a>重新输入!===";
        exit;
    }
    else if($jieshao==""){
        echo "===个人介绍不能为空，请<a href=javascript:history.go(-1)>返回</a>重新输入!===";
        exit;
    }

?>您的自我介绍信息提交成功，请核对!

</div>
<table width="508" border="1" align="center" cellpadding="0" cellspacing="0" bord ercolorlight="#000000" bordercolordark="#FFFFFF" id="table1">
<tr><td width="117" height="22">名字: </td>
```

```
<td width="385" height="22"><?php echo $name?></td>
</tr><tr>
<td height="22">性别：</td><td height="22"><?php echo $sex?></td>
</tr><tr>
<td height="24">年龄：</td><td height="24"><?php echo $age?></td>
</tr><tr>
<td>个人密码：</td><td>已设置</td>
 </tr><tr>
 <td>你的爱好：</td>
     <td>
     <?php echo $like1."  ".$like2."  ".
$like3."  ".$like4."  "?></td>
 </tr><tr>
     <td>你最喜欢的颜色：</td><td><?php echo $color?></td>
</tr><tr>
     <td>个人介绍：</td><td><?php echo $jieshao?></td>
</tr></table>
</body>
```

当表单数据全部接收正确后，其显示效果如图 5-6 所示。

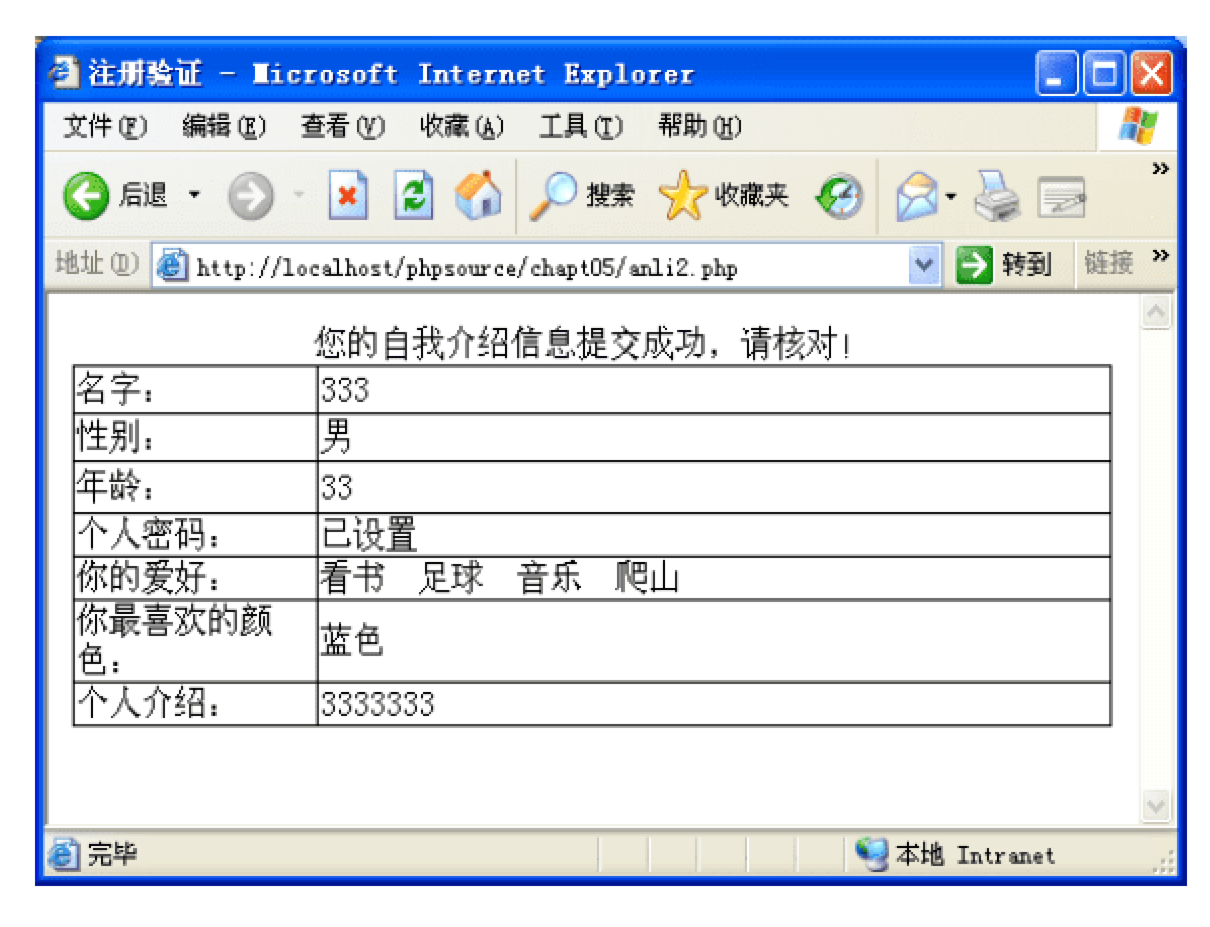


图 5-6 anli2.php 程序的运行结果

5.6 本 章 小 结

本章中先整体介绍了 PHP 5 的内置数组，又通过实例着重讲解了网页间数据传递与接收的方法，用 Cookie 和 Session 实现用户登录的方法以及它们的异同点，用内置数组可以获取哪些服务器信息以及获取方法。通过这一章的的学习，就能设计和制作出很多具有很强交互功能的网页了，比如小型运算器、用户登录、选择题以及猜谜语小游戏等，此外还可以获取服务器的各种信息。

5.7 练 习 题

1. PHP 5 提供了哪些内置数组，各代表什么意思？
2. PHP 5 中实现网页间数据传递，可以使用哪几种方法？
3. PHP 5 中 Cookie 和 Session 都可以实现用户登录，它们有什么区别？
4. PHP 5 中可以通过内置数组获取哪些服务器信息，如何获取？

CHAPTER 6

PHP 5 常用函数介绍

本章重点

- 字符串操作类函数
- 时间日期类函数
- 数学函数
- 图像处理函数

PHP + MySQL

6.1　PHP 5 内置函数概述

6.1.1　标准函数与扩展函数

在第 4 章中已经提到过函数的概念。PHP 中的函数分为内置函数和用户自定义函数两大类。内置函数由 PHP 开发者编写并已嵌入到 PHP 当中，可以在程序中直接使用。而自定义函数则是根据自己的特殊需求编写的函数。实际上 PHP 开发者提供的大量的内置函数已经可以帮助我们轻松完成很多操作。学习和使用函数是学习 PHP 的重要步骤，也是用 PHP 编写复杂程序的重要前提。

PHP 中的内置函数大体分为两大类，一是标准函数库，二是扩展函数库。标准函数库中的函数存放在 PHP 内核中，可以在程序中直接使用，不需要其他任何声明、载入等操作。而扩展函数库中的函数一般不能直接使用，而是按照个人不同的需求来有选择地开启或者关闭。这些扩展函数按照功能的不同被分门别类地封装在多个 DLL 函数库中，这些 DLL 库存放在 PHP 安装文件夹下。在 PHP 5 中，扩展函数被存放在 PHP 安装目录的\ext 子目录下。如图 6-1 所示。

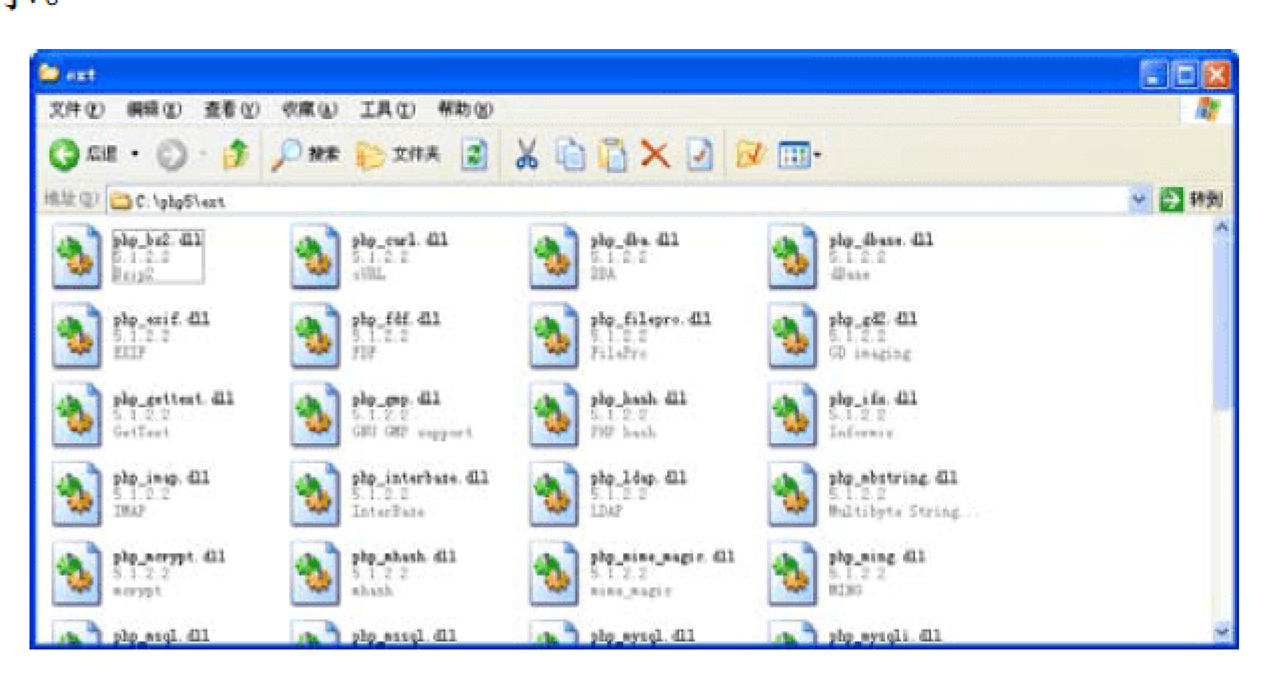


图 6-1　PHP 内置扩展函数库

当需要用到这些扩展函数库中的函数时，只需在 PHP.INI 配置文件中将此扩展库打开即可。它们在 PHP.INI 中的位置如图 6-2 所示。

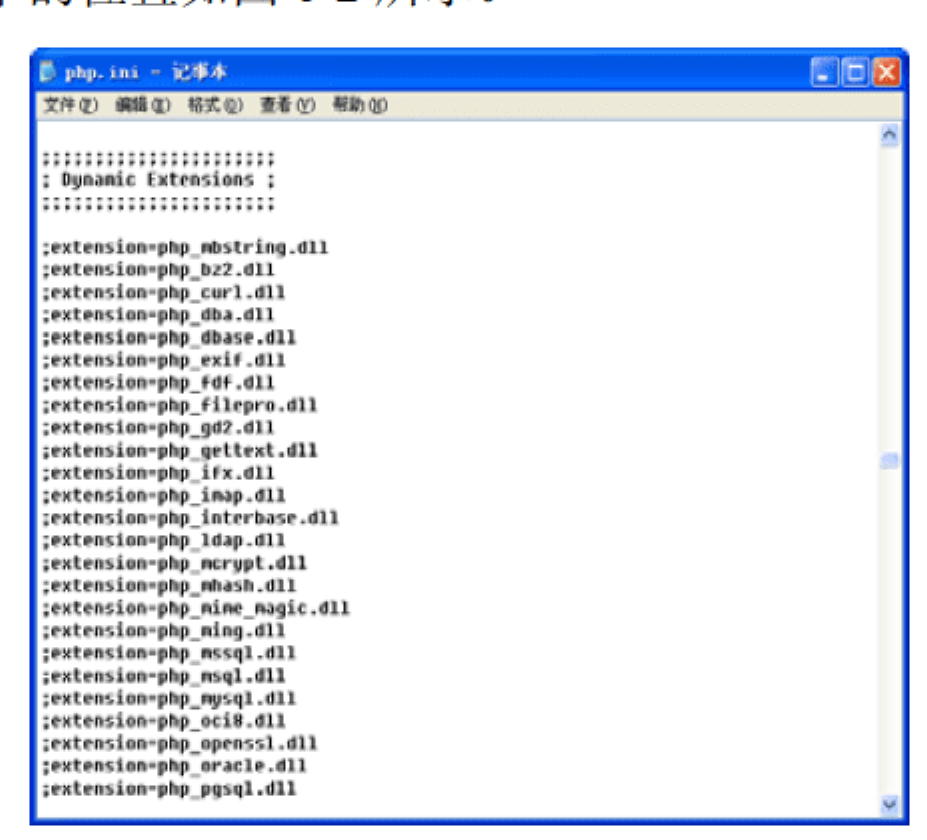


图 6-2　扩展函数库在 php.ini 中的位置

6.1.2 启用扩展函数库

打开一个扩展库的方法很简单，只需要将“;extension=php_xxx.dll”前面的分号“;”去掉，保存 php.ini，然后重新启动 IIS 或者 Apache，此时 php.ini 生效，此扩展随即可以使用。

在本书的程序中用到的函数，绝大多数都属于标准内置函数，不需要对 PHP 做特殊的配置就能够直接使用。如果用到扩展函数库中的函数，我们会及时说明。读者如果在编写程序过程中，遇到有的 PHP 函数不能使用的情况，应当考虑是否是因为没有打开相应的扩展库。如果没有打开相应的扩展库，PHP 系统会给出“Fatal error:Call to undefined function…”这样的提示。

接下来将在本章中介绍一些最为常用的 PHP 内置函数。

6.2 数 组 函 数

6.2.1 数组函数一览

数组是编程中的重要元素，在 PHP 中也不例外。在 PHP 语法部分已经介绍了数组的定义和使用，对数组有了一定了解。PHP 中还提供了一些用来操作数组的函数，这些函数作为标准函数可以直接使用。表 6-1 列出了 PHP 5.1.6 提供的数组函数。

表 6-1 PHP 5 的数组操作函数

函数名	函数功能概要
array_change_key_case	返回字符串键名全为小写或大写的数组
array_chunk	将一个数组分割成多个
array_combine	创建一个数组，用一个数组的值作为其键名，另一个数组的值作为其值
array_count_values	统计数组中所有的值出现的次数
array_fill	用给定的值填充数组
array_flip	交换数组中的键和值
array_keys	返回数组中所有的键名
array_map	将回调函数作用到给定数组的单元上
array_merge_recursive	递归地合并一个或多个数组
array_merge	合并一个或多个数组
array_multisort	对多个数组或多维数组进行排序
array_pad	用值将数组填补到指定长度
array_pop	将数组最后一个单元弹出（出栈）

（续表）

函数名	函数功能概要
array_product	计算数组中所有值的乘积
array_push	将一个或多个单元压入数组的末尾（入栈）
array_rand	从数组中随机取出一个或多个单元
array_reverse	返回一个单元顺序相反的数组
array_shift	将数组开头的单元移出数组
array_slice	从数组中取出一段
array_splice	把数组中的一部分去掉并用其他值取代
array_sum	计算数组中所有值的和
array_unique	移除数组中重复的值
array_unshift	在数组开头插入一个或多个单元
array_values	返回数组中所有的值
array_walk_recursive	对数组中的每个成员递归地应用用户函数
array_walk	对数组中的每个成员应用用户函数
array	新建一个数组
arsort	对数组进行逆向排序并保持索引关系
asort	对数组进行排序并保持索引关系
compact	建立一个数组，包括变量名和他们的值
count	计算数组中的单元数目或对象中的属性个数
current	返回数组中的当前单元
each	返回数组中当前的键 / 值对并将数组指针向前移动一步
end	将数组的内部指针指向最后一个单元
extract	从数组中将变量导入到当前的符号表
in_array	检查数组中是否存在某个值
key	从关联数组中取得键名
krsort	对数组按照键名逆向排序
ksort	对数组按照键名排序
list	把数组中的值赋给一些变量
natcasesort	用“自然排序”算法对数组进行不区分大小写字母的排序
natsort	用“自然排序”算法对数组排序
next	将数组中的内部指针向前移动一位
prev	将数组的内部指针倒回一位
range	建立一个包含指定范围单元的数组

（续表）

函数名	函数功能概要
reset	将数组的内部指针指向第一个单元
rsort	对数组逆向排序
shuffle	将数组打乱
sizeof	count()的别名
sort	对数组排序
uasort	使用用户自定义的比较函数对数组中的值进行排序并保持索引关联
uksort	使用用户自定义的比较函数对数组中的键名进行排序
usort	使用用户自定义的比较函数对数组中的值进行排序

看到这个表读者可能大吃一惊，PHP 开发者竟然为我们提供了如此多的函数！实际上，PHP 提供的数组操作函数多达 110 多个。表 6-1 中并未列出其全部，而只列出了其中较为常用的部分。

有些读者可能会被这密密麻麻的函数吓倒，认为学习 PHP 很难。实际上每一个学习者都不可能把这所有的函数都记住。除了少量的极为常用函数最好记住以外，其他大多数函数都没有必要去死记硬背。一种比较好的学习方法是将所有函数浏览一遍，并大体记住其功能。等到编程中遇到相似问题时，可以通过查找函数手册找到函数的使用方法，然后应用到程序当中去。实际上很多编程语言的函数库、类库都很庞大，不可能一下子掌握，都有一个逐渐熟悉、积累的过程。

下面以表中列出的最常用的几个函数为例来说明数组处理函数的使用，其他函数读者可以参考 PHP 手册来学习其使用方法。

6.2.2 array()函数

array()函数用来建立一个新数组。函数的参数可以是混合类型。看一个例子：

程序清单 文档 6-1.php

```
<HTML>
<HEAD>
<TITLE>Array 函数的使用</TITLE>
</HEAD>
<BODY>
<?php
$arr1=array(0,1,2,3,4);
$arr2=array("a"=>0,"b"=>1,"c"=>2,"d"=>3,"e"=>4);
echo "\$arr1[0]=".$arr1[0];
echo "<br>";
echo "\$arr2[\"a\"]=".$arr2["a"];
?>
```

```
</BODY>
</HTML>
```

上例中，首先用 array()函数定义了拥有 5 个元素的数组$arr1，每个元素分别赋值 0，1，2，3，4。然后定义同样拥有 5 个元素的数组$arr2，并分别赋值 0，1，2，3，4。两个数组的不同是第一个数组用了默认的数字作为下标，第二个数组用了自定义的字符作为下标。最后输出数组元素时也使用了各自对应的下标。程序的运行结果如图 6-3 所示。

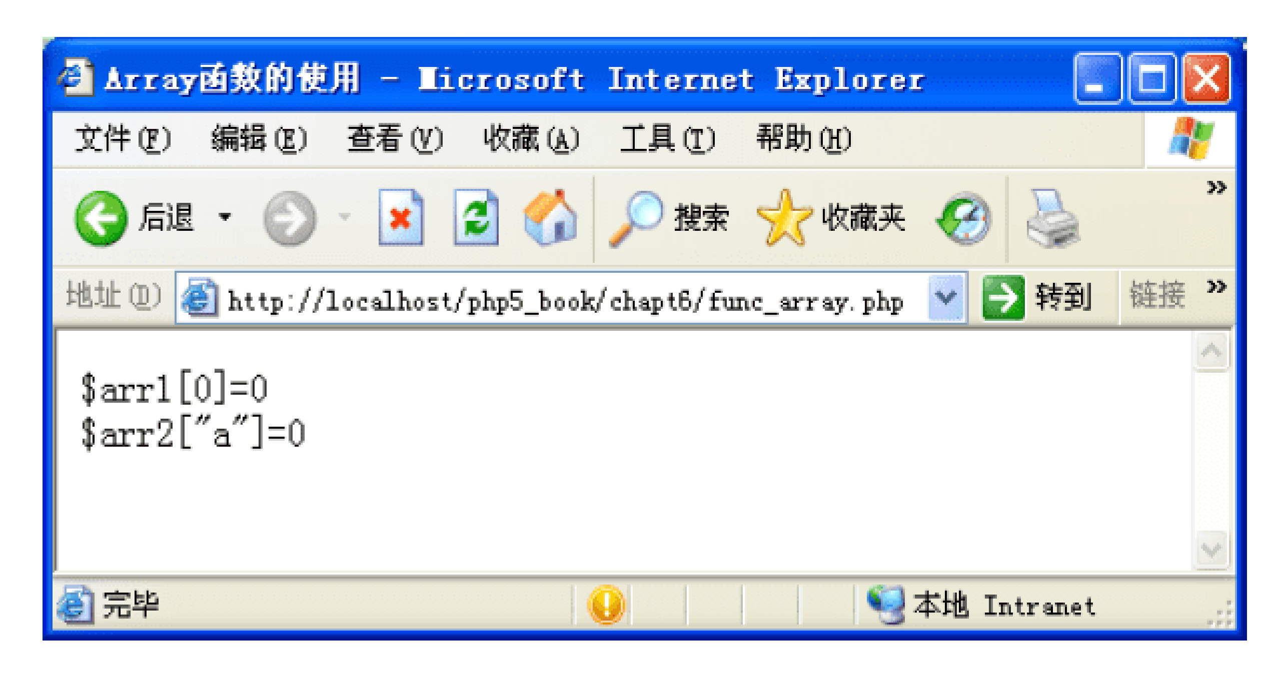


图 6-3　array()函数的使用

6.2.3　count()函数

count()函数可以用来统计一个数组中元素的个数，这在循环遍历一个未知长度的数组时尤其有用。看下面的例子：

程序清单　文档 6-2.php

```
<HTML>
<HEAD>
<TITLE>Count 函数的使用</TITLE>
</HEAD>
<BODY>
<?php
$arr1=array(0,1,2,3,4);
Echo "数组\$arr1 中元素的个数为：".count($arr1);
?>
</BODY>
</HTML>
```

程序运行后将输出："数组$arr1 中元素个数为：5"。

6.2.4　each()函数

each()函数可以返回一个数组中当前元素的键和值，并将数组指针向前移动一步。它常常被用在一个循环中，遍历一个数组。看下面的例子：

程序清单　文档 6-3.php

```
<HTML>
<HEAD>
<TITLE>each 函数的使用</TITLE>
</HEAD>
<BODY>
<?php
$arr=array("name"=>"Bob","age"=>20,"sex"=>"male","postcode"=>"100000");
For($i=0;$i<count($arr);$i++){
$kav=Each($arr);
Echo $kav["key"]."=>".$kav["value"];
echo "<br>";
}
?>
</BODY>
</HTML>
```

上面的程序中，首先定义了一个数组$arr，并且为其赋值。值得注意的是数组下标不是按顺序递增的数字，而是毫无规律的字符串。所以不能直接用一个递增的数字作为下标来输出，循环输出遇到了困难。但是使用了 each()函数之后，可以获得这个数组的下标以及下标对应的值，因此就可以动态循环输出每一个元素的下标和值。each($arr)这句代码将$arr 数组中当前元素的下标和值都存放到另外一个数组$kav 中，然后将数组指针指到下一个元素。并且$kav 数组的下标分别为 key 和 value。这样只需要调用$kav[“key”]和$kav[“value”]即可获得数组$kav 当前元素的下标和值。输出这两个值后当前循环结束，执行下一次循环，这样又输出了下一个元素的值，依此类推，整个数组都被动态循环输出了。程序的运行结果如图 6-4 所示。

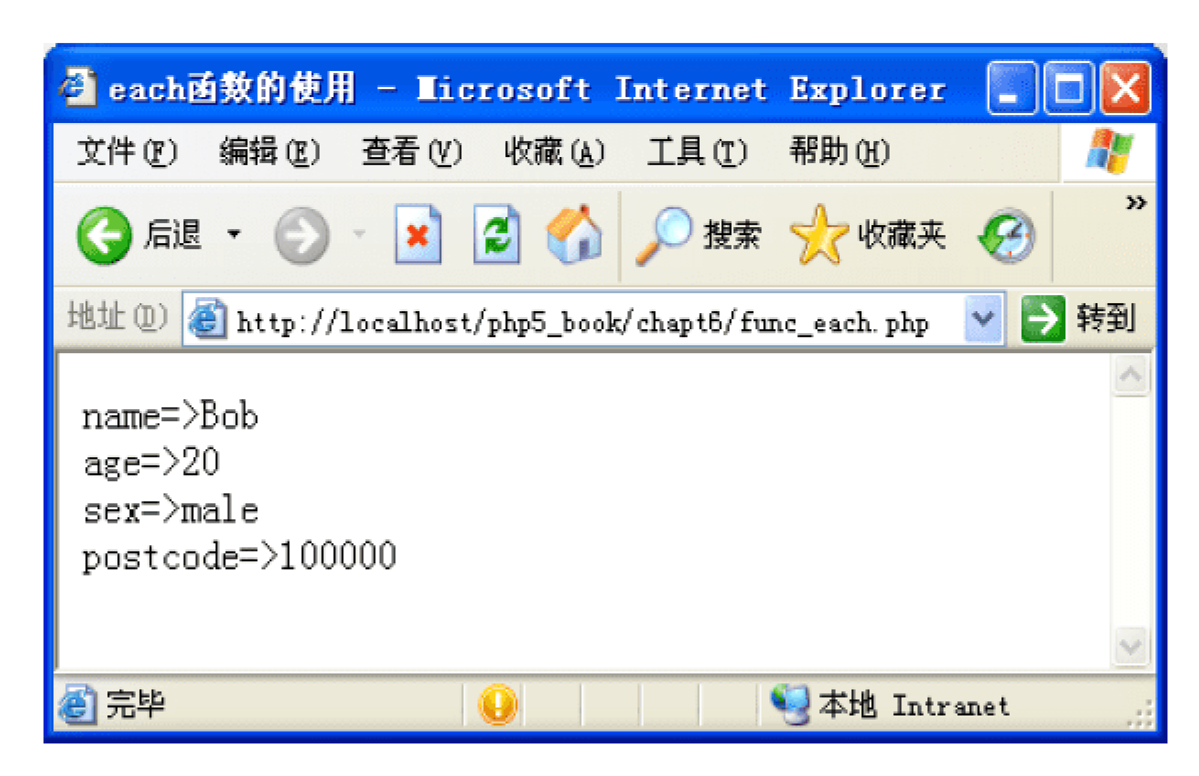


图 6-4　使用 each()函数循环输出数组元素

通过上面的例子已经可以看出 each()函数的奇妙作用。其实例 6-3 中的代码还可以继续简化，甚至可以不用 count()函数来统计数组元素个数，也可以实现动态循环输出一个未知长度的数组。看下面的例子：

程序清单　文档 6-4.php

```
<HTML>
<HEAD>
<TITLE>EACH 函数应用范例 2</TITLE>
</HEAD>
<BODY>
<?php
$arr=array("name"=>"Bob","age"=>20,"sex"=>"male","postcode"=>"100000");
While($kav=each($arr)){
Echo $kav["key"]."=>".$kav["value"];
echo "<br>";
}
?>
</BODY>
</HTML>
```

例 6-4 的代码比例 6-3 更加简洁，实现的效果却完全相同。例 6-4 利用了 each()函数一个重要性质，那就是当数组已经到达末尾时 each()函数返回 false。我们已经知道 false 是一个布尔值，表示“否”，因此它正好可以作为 while 循环的结束条件。这样，用一个 while 循环来读取$arr 数组中的每一个元素，不管数组有多少个元素，当指针到达末尾时，each($arr)返回 false，循环结束，程序执行完成。同样实现了动态输出未知长度的数组的功能。

通过例 6-4 也可以说明，有时候实现同一个功能，有很多途径可以选择。作为程序开发人员，应该尽量选择更加简洁、高效的途径。

6.2.5　current()、reset()、end()、next()和 prev()函数

之所以要将这 5 个函数并列起来介绍，是因为这 5 个函数的作用相似——它们都用来进行数组内部的指针操作。在 PHP 中，使用一个内部指针来指向一个数组。当需要数组中的某一元素时，只需要将指针移动到该元素的位置，即可取出该元素。这大大方便了对数组的操作。先详细说明一下这 5 个函数的作用，然后通过一个例子来验证其使用效果。

- current()：返回当前内部指针所指的元素的值。当到达数组末尾时返回 false。
- reset()：将内部指针指向数组的第一个元素，并返回其值。数组为空时返回 false。
- end()：将内部指针指向数组的最后一个元素，并返回其值。
- next()：将数组指针指向当前元素的下一个元素，并返回其值。到达末尾时返回 false。
- prev()：将数组指针指向当前元素的上一个元素，并返回其值，当到达顶端时返回 false。

上面 5 个函数的返回值均为 mixed 类型，根据数组的元素值类型不同而返回不同的类型。在这里要注意 current()函数和 next()函数的不同，它们虽然都是取出一个元素值，但是 current()并不移动指针。也就是说 current()返回的是未移动指针之前所指向的元素的值，而 next()返回的是移动指针之后所指向的元素的值。下面来看一个例子：

程序清单 文档 6-5.php

```
<HTML>
<HEAD>
<TITLE>数组内部指针移动函数的综合应用</TITLE>
</HEAD>
<BODY>
<?php
$arr=array(1,2,3,4,5,6,7,8,9,10);
Echo "调用 current():".current($arr);
echo "<br>";
echo "再次调用 current():".current($arr);
echo "<br>";
echo "调用 next():".next($arr);
echo "<br>";
echo "调用 reset():".reset($arr);
echo "<br>";
echo "调用 end():".end($arr);
echo "<br>";
echo "调用 prev():".prev($arr);
?>
</BODY>
</HTML>
```

本例中，定义了一个数组$arr，并且用 10 个数字对其作了初始化。然后分别调用上述 5 个函数来观察其运行效果。为使输出结果直观，在每一次调用之后都输出一个换行。程序运行结果如图 6-5 所示。

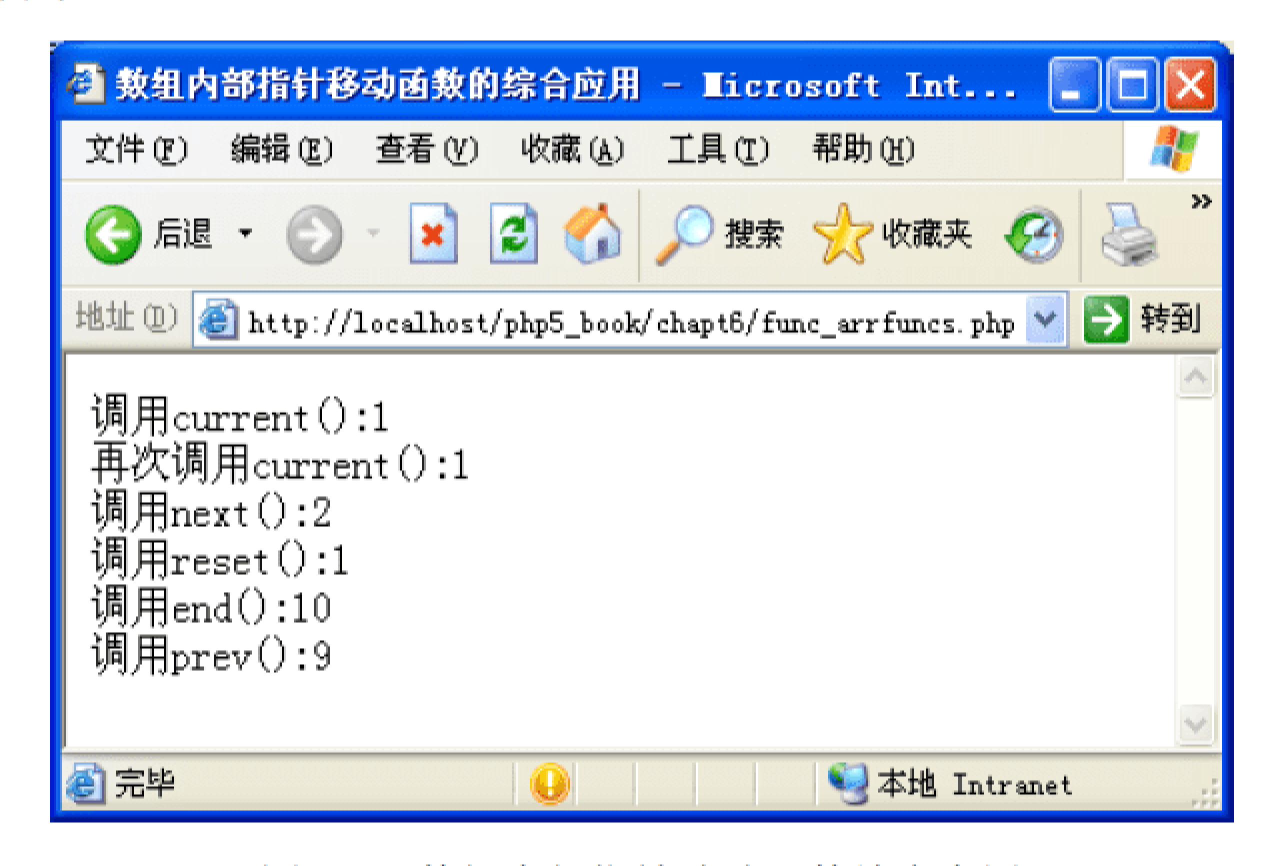


图 6-5 数组内部指针移动函数综合应用

下面分析一下程序的运行流程和对应的输出结果：

（1）数组初始化完成，内部指针指向第一个元素（元素值为 1）。

（2）第一次调用 current()函数，返回当前元素值 1，指针不变。

（3）再次调用 current()函数，由于内部指针不变，仍然返回 1。

（4）调用 next()函数，内部指针指向下一个元素，并返回其值（返回 2）。

（5）调用 reset()函数，内部指针再次指向第一个元素，返回 1。

（6）调用 end()函数，内部指针指向最后一个元素，并返回其值（返回 10）。

（7）调用 prev()函数，内部指针指向前一个元素，并返回其值（返回 9）。

关于 PHP 的数组函数，就介绍到这里。表 6-1 中列出的其他函数，如果读者感兴趣可以自行编写程序试验一下。函数的参数、返回值类型等均可以通过查看 PHP 手册获得。

6.3　字符串处理函数

6.3.1　字符串处理函数一览

在 Web 编程中，字符串是使用最为频繁的数据类型之一。此外 PHP 不是一门强固类型化语言，因此很多数据都可以作为字符串来方便处理。字符串操作是编程中极为常用的操作，从简单的打印输出一行字符串到复杂的正则表达式等，处理对象都是字符串。PHP 为我们提供了大量实用的函数，帮助我们完成许多复杂的字符串处理工作。函数列表如表 6-2 所示。

表 6-2　PHP 常用字符串操作函数一览

函数名	函数功能概要
addcslashes	像 C 那样使用反斜线转义字符串中的字符
addslashes	使用反斜线引用字符串
bin2hex	将二进制数据转换成十六进制表示
chop	rtrim()的别名
chr	返回指定的字符
chunk_split	将字符串分割成小块
convert_cyr_string	将字符由一种 Cyrillic 字符转换成另一种
convert_uudecode	对一个未编码字符串进行编码
convert_uuencode	对一个字符串进行解码
count_chars	返回字符串所用字符的信息
crc32	计算一个字符串的 crc32 多项式
echo	输出字符串
explode	使用一个字符串分割另一个字符串
fprintf	将格式化字符串写入流
html_entity_decode	将 HTML 标记转换为特殊字符
htmlspecialchars	将特殊字符转换为 HTML 标记

（续表）

函数名	函数功能概要
implode	合并数组元素到一个字符串中
join	implode 函数的别名
ltrim	去除字符串左侧空格
md5_file	用 md5 算法对文件进行加密
md5	用 md5 算法对字符串进行加密
nl2br	将换行符替换成 HTML 的换行符
number_format	将一个数字格式化成三位一组
ord	返回一个字符的 ASCII 码
print	输出字符串
printf	输出格式字符串
rtrim	去除字符串右侧空格
sprintf	返回一个格式字符串
str_pad	用一个字符串填充另外一个字符串到一定长度
str_repeat	重复输出一个字符串
str_replace	字符串替换
str_shuffle	随机打乱一个字符串
str_split	将字符串转换成数组
str_word_count	统计字符串中的单词数
strchr	strstr()的别名
strcmp	字符串比较大小
strip_tags	过滤掉字符串中的 PHP 和 HTML 标记
strlen	获得字符串的长度（所占字节数）
strpbrk	以子串中的任意一个字符第一次出现的位置为界将字符串分成两部分，并返回后半部分
strpos	查找一个子串在字符串中第一次出现的位置
strrpos	查找一个子串在字符串中最后一次出现的位置
strrev	将一个字符串顺序倒转
strrchr	查找一个字符在一个字符串中最后一次出现的位置并返回从此位置开始之后的字符串
strstr	查找一个子串在一个字符串中第一次出现的位置，并返回从此位置开始的字符串
strstr	strrchr 函数的别名
strtok	将字符串打碎成小段
strtolower	将字符串中的字符全部小写
strtoupper	将字符串中的字符全部大写
strtr	批量字符替换
substr_count	统计一个子串在字符串中的出现次数

（续表）

函数名	函数功能概要
substr_replace	在字符串内部制定区域内替换文本
substr	截取字符串的一部分
trim	去除字符串首尾的连续空格
ucfirst	将字符串首字母大写
ucwords	将字符串中每个单词的首字母大写

表 6-2 列出了常用的 PHP 字符串操作函数。接下来我们选最常用的进行介绍。

6.3.2　trim()、ltrim()、rtrim()、chop()和 strlen()函数

这 5 个函数中前 4 个函数的功能类似，因此将其放在一起介绍。其中 chop()函数与 rtrim()函数作用相同，都是去除字符串右端的空格。ltrim()用来去除字符串左端的空格，而 trim()用来去除字符串左右两端的空格。

下面来看一个例子。例子中还用到了另外一个字符串处理函数 strlen()来获得字符串的长度。

程序清单　文档 6-6.php

```
<!—文件 6-6.php：去除字符串两端空格-->
<HTML>
<HEAD>
<TITLE>去除字符串两端空格</TITLE>
</HEAD>
<BDOY>
<?php
$str="  你看不到我 我是空格 ";
echo "方括号中为原始字符串：[".$str."]<br>";
echo "原始字符串长度：".strlen($str)."<br>";
$str1=ltrim($str);
echo "执行 ltrim()之后的长度：".strlen($str1)."<br>";
$str2=rtrim($str);
echo "执行 rtrim()之后的长度：".strlen($str2)."<br>";
$str3=trim($str);
echo "执行 trim()之后的长度".strlen($str3)."<br>";
echo "去掉首尾空格之后的字符串：[".$str3."]";
?>
</BDOY>
</HTML>
```

程序的运行结果如图 6-6 所示。

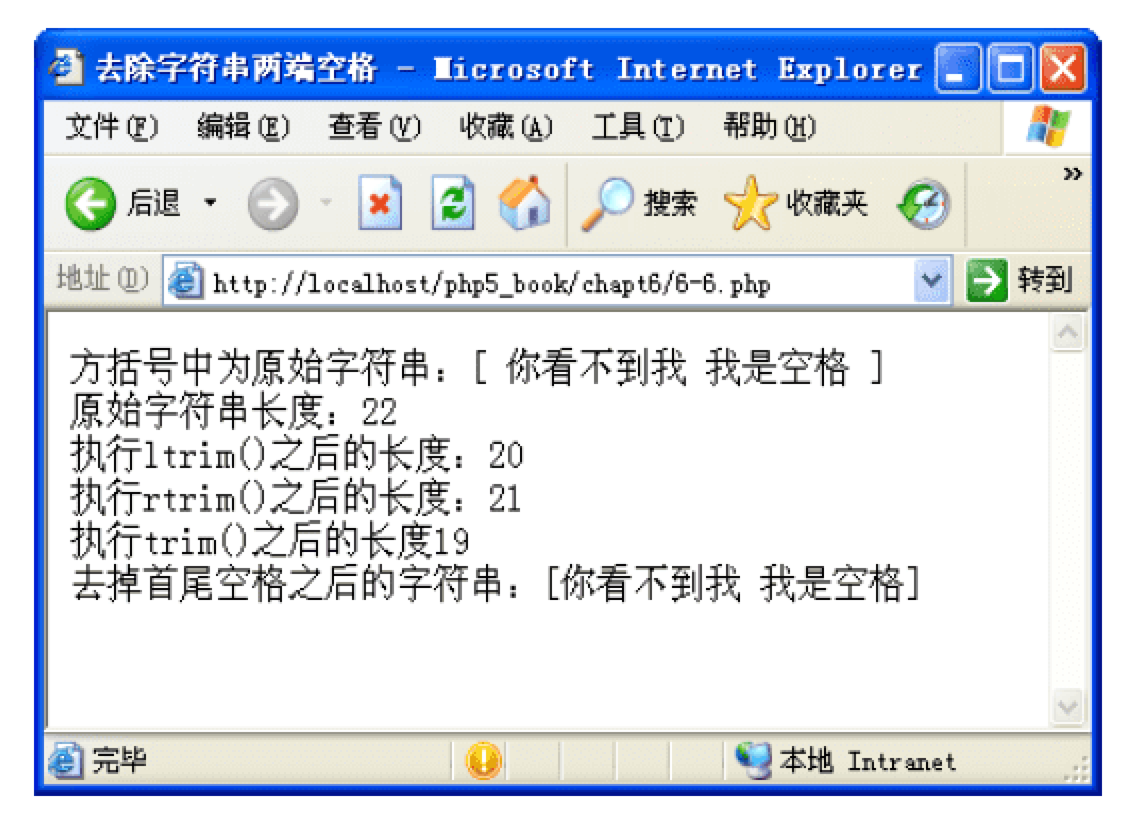


图 6-6　6-6.php 运行结果

在本程序中，首先构造了一个字符串变量$str。这个字符串由 9 个汉字和 4 个空格组成。并且 4 个空格中有 2 个在左侧，1 个在中间，1 个在右侧（由于浏览器会忽略掉连续的空格，因此在浏览器中连续的两个空格的显示效果与一个空格相同）。由于每个汉字占 2 个字节，每个英文半角空格占 1 个字节，因此初始字符串的长度为 9*2+4=22。用 strlen()函数来输出其长度。

首先执行 ltrim()函数，将返回结果存放在$str1 中。由于 ltrim()函数将会去掉字符串左侧的所有连续的空格，因此两个空格就被去掉，因此$str2 的字符串长度就是 20。

然后执行 rtrim()函数，将返回结果存放在$str2 中。rtrim()函数去掉了字符串$str 的右侧一个空格，因此$str2 的长度为 21。

最后执行 trim()函数。trim()函数去除字符串左右两侧的所有空格，因此左侧的 2 个空格和右侧的 1 个空格被去掉，剩余的部分长度为 19。通过$str3 的输出也可以看出，字符串两侧的空格已经消失。

去除连续的空格往往被用在作字符串比较之前。比较两个字符串是否相同，如果其中一个字符串首尾带有空格，那比较结果就会为假。如“abc”和“abc ”这两个字符串，看似内容完全相同，但由于后者后面多了一个空格，在比较时会返回 false。因此在比较两个字符串变量值是否相同时往往首先用 trim()函数处理一下两侧的空格。

值得注意的是，trim()函数只去除字符串两侧的空格，而不会去除中间的空格。如例 6-6 中，“你看不到我”和“我是空格”之间有一个空格。调用这三个函数之后空格仍然存在，字符串中间的空格不受影响。如果想去除掉一个字符串中的所有空格，可以使用后面要讲的字符串替换函数来实现。

6.3.3　ucwords()、ucfirst()、strtoupper()、strtolower()、str_word_count()函数

这 5 个函数提供了对字符串中的单词的处理。包括转换大小写、计算单词个数等。下面还是通过一个例子来了解。

程序清单　文档 6-7.php

```
<!--文件 6-7.php：字符串处理中的单词处理-->
<HTML>
<HEAD>
<TITLE>字符串处理中的单词处理</TITLE>
</HEAD>
<BDOY>
<?php
$str="ni hao,wo jiao Wang Xiao-ming.";
echo "原始字符串：".$str."<br>";
$str1=ucfirst($str);
echo "执行 ucfirst()之后：".$str1."<br>";
$str2=ucwords($str);
echo "执行 ucwords()之后：".$str2."<br>";
$str3=strtoupper($str);
echo "执行 strtoupper()之后：".$str3."<br>";
$str4=strtolower($str);
echo "执行 strtolower()之后：".$str4."<br>";
echo "字符串中一共有：".str_word_count($str)."个单词。";
?>
</BDOY>
</HTML>
```

本例构造了一个包含有 6 个单词，并且大小写混合的字符串，来测试函数的运行结果。首先调用 ucfirst()函数来将整个字符串首字母变为大写，然后用 ucwords()函数将每个单词的首字母变为大写，然后调用 strtoupper()函数将全部字母都变成大写，调用 strtolower()函数将所有字母变成小写，最后调用 str_word_count()函数统计字符串中的单词个数。程序的运行结果如图 6-7 所示。

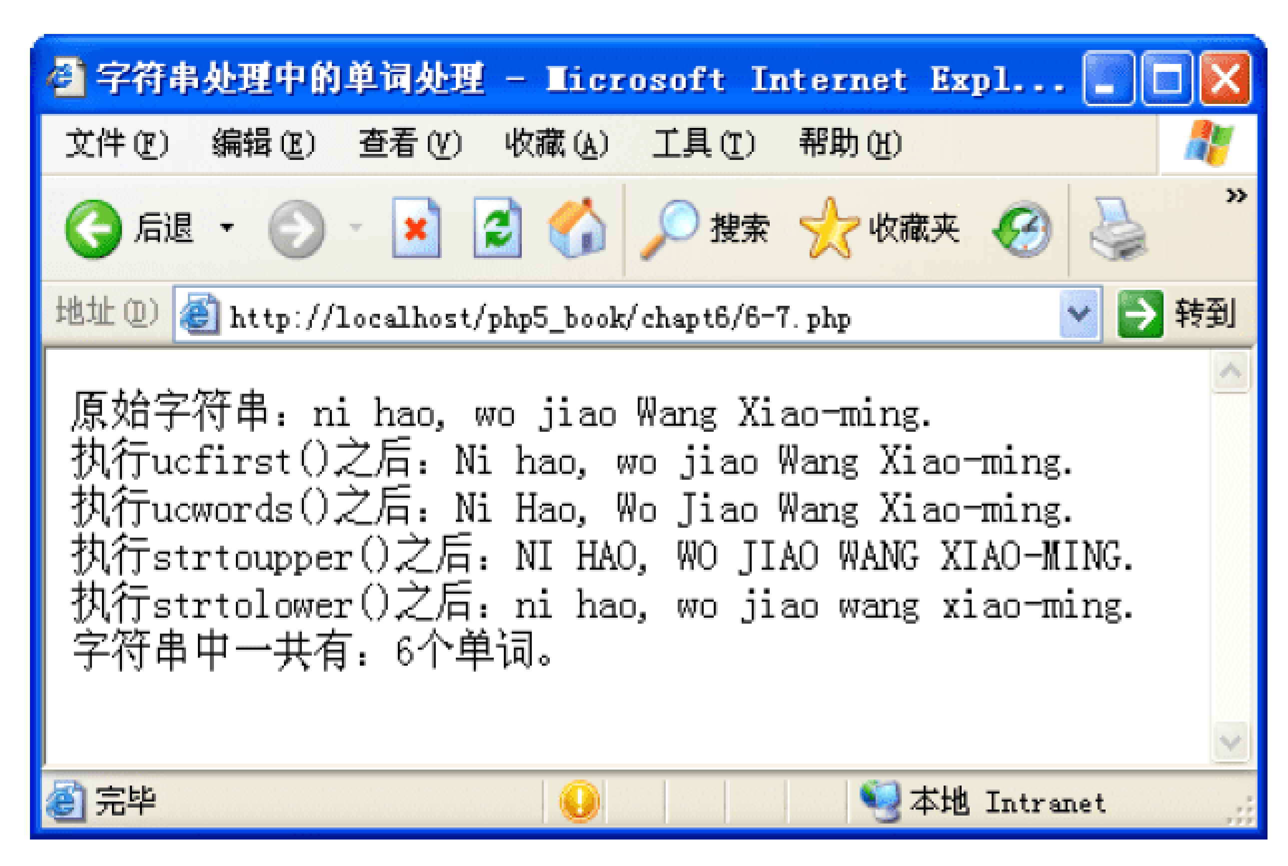


图 6-7　使用字符串处理函数对单词进行处理

6.3.4 字符串查找函数

我们经常要用到在一个字符串中查找某个字符或者某个子串的操作，以及对字符串中的某些字符按照需求进行替换操作，或是截取字符串的一部分等等。对于这些操作，PHP 都有一系列函数实现，我们只需要了解函数的使用方法，即可轻松实现。

常用的字符串查找函数有 substr_count()、strpos()、strrpos()、strstr()、strrchr()等。它们的使用方法和功能介绍如下。

1. substr_count()函数

substr_count()函数的格式为：

```
int substr_count(string haystack,string needle [,int offset [,int length]])
```

上述语句中，substr_count()用来统计一个字符串 needle 在另一个字符串 haystack 中出现的次数，该函数返回值是一个整数。有两个可选参数 offset 和 length，分别表示要查找的起点和长度。值得注意的是，offset 是从 0 而不是从 1 开始计算的。比如下面的代码：

程序清单 文档 6-8.php

```
<!--代码 6-8：substr_count()函数统计字符串出现次数-->
<HTML>
<HEAD>
<TITLE>用 substr_count 函数统计字符串出现次数</TITLE>
</HEAD>
<BDOY>
<?php
$str="I am an abstract about abroad.";
echo substr_count($str,"ab");
echo ", ";
echo substr_count($str,"ab",6,4);
?>
</BDOY>
</HTML>
```

本例的输出结果为“3，1”。

substr_count($str,“ab”)的作用是返回字符串“ab”在字符串$str 中出现的次数，由于“ab”在整个字符串中出现了三次，因此值为 3。

substr_count($str,“ab”,6,4)的作用是返回字符串“ab”在$str 中的从第 6 个字符开始（包含第 6 个字符，而且从 0 开始计算），往后数 4 个字符为止（即第 9 个字符）之间的字符串中出现的次数。也就是参数“6，4”限定了查找字符串的范围。不指定参数时，substr_count 函数从整个字符串$str 中查找“ab”的出现次数，而指定了参数之后，substr_count 函数从指定的范围内查找“ab”的出现次数。这个范围就是从字符串的第 6 个字符开始到第 9 个字符为止的 4 个字符。具体到本例中，就是“n ab”（注意 n 和 a 之间的一个空格也算一个字符）。在这个范围内，“ab”只出现了一次，因此返回 1。

2. strrpos()函数和 strpos()函数

strrpos()函数的格式为：

```
int strrpos(string haystack,mixed needle [,int offset])
```

strrpos()函数返回字符 needle 在字符串 haystack 中最后一次出现的位置。这里 needle 只能是一个字符，而不能是一个字符串。如果提供一个字符串，PHP 也只会取字符串的第一个字符使用，其他字符无效。参数 offset 也是用来限制查找的范围。

strpos()函数的格式为：

```
int strpos(string haystack,mixed needle [,int offset])
```

strpos()函数与 strrpos()函数仅一字之差，但功能相差很大。strpos()函数的 needle 参数允许使用一个字符串，而且返回的是这个字符串在 haystack 中第一次出现的位置，而不是最后一次。下面来看一个例子。

程序清单　文档 6-9.php

```
<--文件 6-9.php：在字符串中查找一个字串-->
<HTML>
<HEAD>
<TITLE>字符串查找函数的使用（一）</TITLE>
</HEAD>
<BDOY>
<?php
$str="I am an abstract about abroad.";
echo "原始字符串为：".$str."<br>";
echo "ab 在字符串中的第一次出现位置为：".strpos($str,"ab")."<br>";
echo "ab 在字符串中的最后一次出现位置为：".strrpos($str,"ab")."<br>";
echo "abcd 在字符串中第一次出现的位置为：".strpos($str,"abcd");
?>
</BDOY>
</HTML>
```

本程序的运行结果如图 6-8 所示。

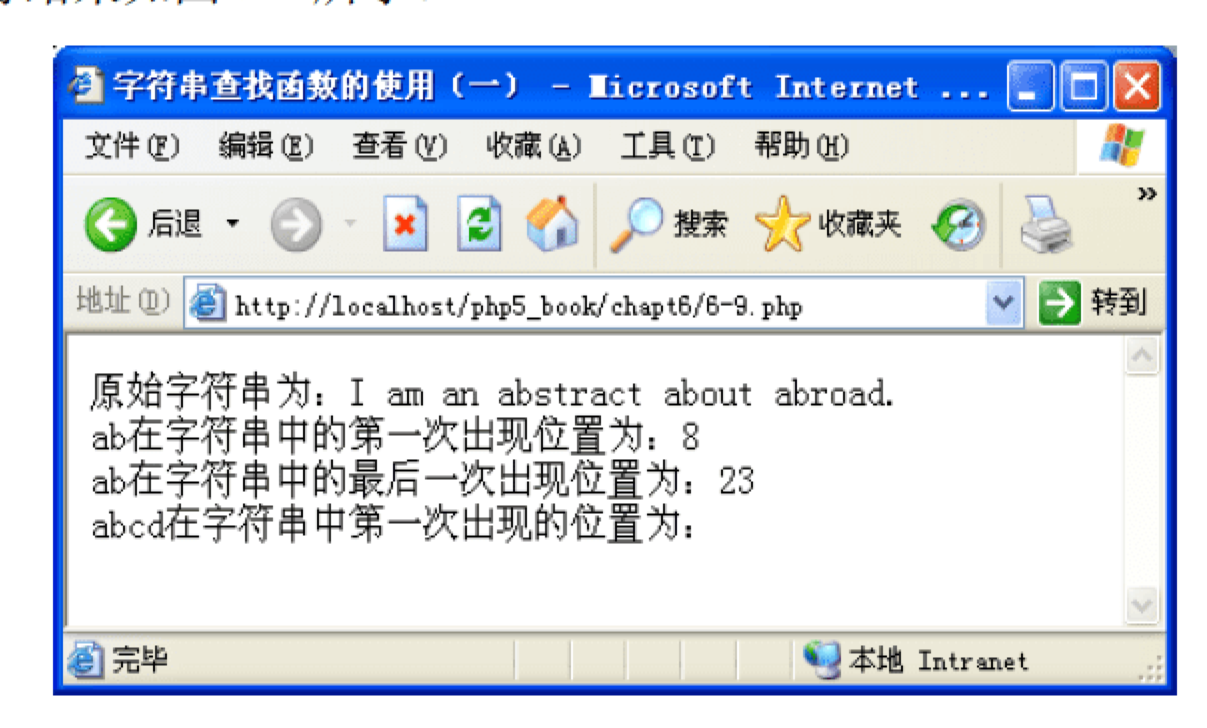


图 6-8　在字符串中查找子串

本程序中首先构造了一个包含多个“ab”的字符串。然后分别调用 strpos()和 strrpos()函数来获得“ab”子串在字符串中第一次和最后一次出现的位置。程序分别给出了输出结果 8 和 23。这里有两点值得注意：一是这里的 8 和 23 都是从 0 开始计算，而且是从子串的第一个字母出现的位置开始计算。如子串为“ab”，那找到 ab 之后，以“a”的位置序号作为函数的返回值，而不是 b 的位置序号；二是如果要查找的字符串不存在，则返回布尔值 false，由于 false 无法直接输出，因此最后查找“abcd”子串时没有任何输出。

3. strstr()函数和 strrchr()函数

strstr()函数和 strrchr()两个函数的格式分别是：

```
string strstr(string haystack,string needle)
string strrchr(string haystack,string needle)
```

由此可见，这两个函数均返回一个字符串，而不是返回一个表示位置的整数。两个函数除了函数名不同，使用方法完全相同，但是其作用略有不同。strstr()函数用来查找一个子串 needle 在一个字符串 haystack 中第一次出现的位置，并返回从此位置开始的字符串。strrchr()函数查找一个字符 needle 在一个字符串 haystack 中最后一次出现的位置，并返回从此位置开始之后的字符串。用一个例子来具体说明：

程序清单　文档 6-10.php

```
<!--代码 6-10：strstr 和 strrchr 函数使用范例-->
<HTML>
<HEAD>
<TITLE>字符串查找函数的使用（二）</TITLE>
</HEAD>
<BDOY>
<?php
$str="千山鸟飞绝，万径人踪灭，孤舟蓑笠翁，独钓寒江雪。";
echo "1.原始字符串为：".$str."<br>";
echo "用 strstr 函数搜索‚，‛的返回结果：".strstr($str,"，")."<br>";
echo "用 strstr 函数搜索‚孤舟‛的返回结果：".strstr($str,"孤舟")."<br>";
$str2="I have a great dream.";
echo "2.原始字符串为：".$str2."<br>";
echo "用 strrchr 函数搜索"e"的返回结果：".strrchr($str2,"e")."<br>";
echo "试图用 strrchr 函数搜索"at"的返回结果：".strrchr($str2,"at");
?>
</BDOY>
</HTML>
```

程序的运行结果如图 6-9 所示。

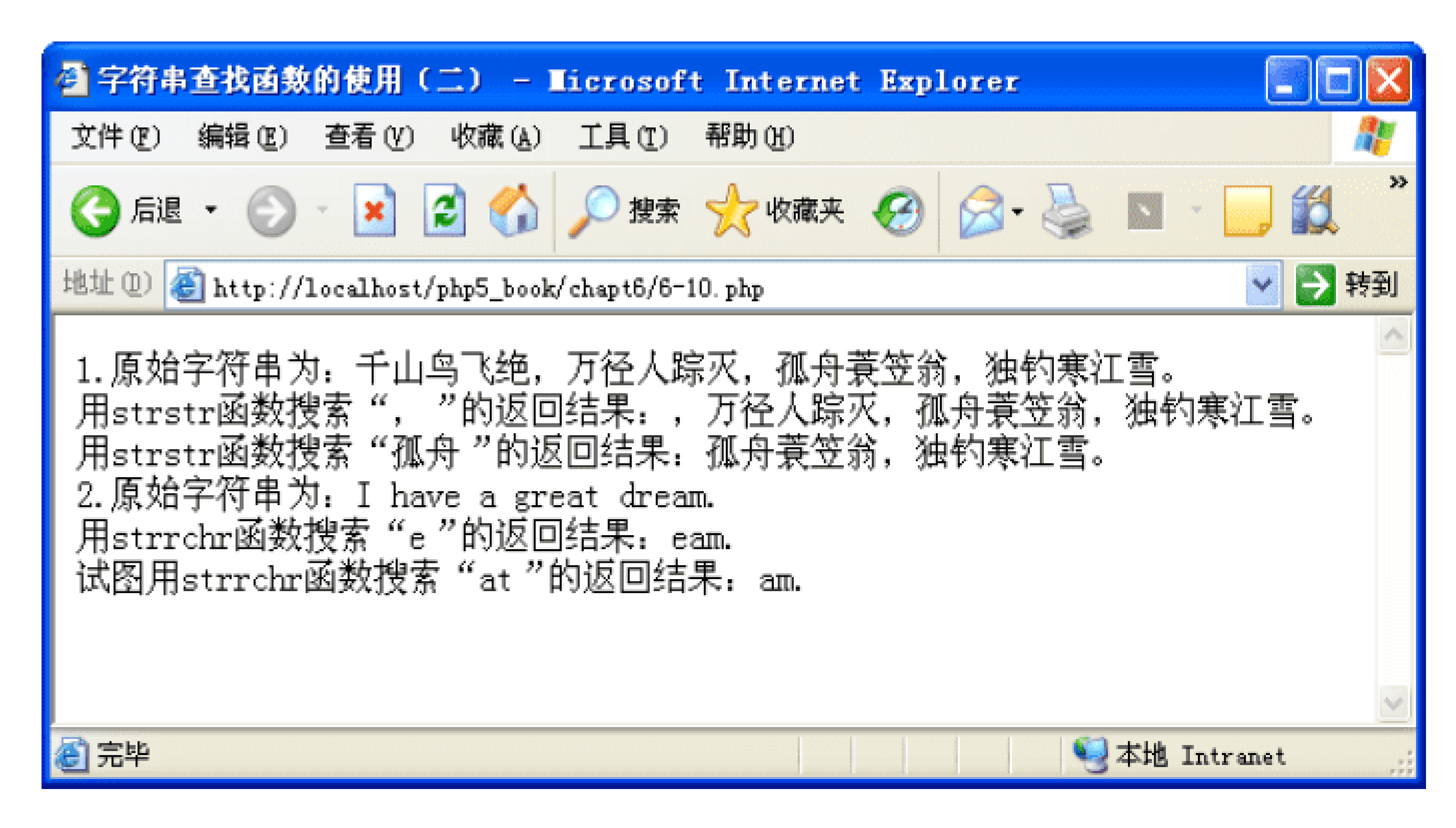


图 6-9　strstr()函数和 strrchr()函数的使用

分析本例的输出结果，就能够准确的把握 strstr()和 strrchr()这两个函数的功能特点。

首先，在第一个字符串中，用 strstr 函数搜索逗号“，”。该函数返回字符串中第一次出现“，”的位置之后的字符串。第一次出现逗号是在“千山鸟飞绝”的“绝”字之后，因此，函数的返回结果就是“，万径人踪灭，孤舟蓑笠翁，独钓寒江雪。”（注意逗号本身也会被返回）。为了证明 strstr()函数可以使用一个字符串而不仅仅是单个字符作为参数，我们又在字符串中搜索“孤舟”。按照上面对 strstr()函数的了解，函数应当返回“孤舟蓑笠翁，独钓寒江雪。”，和程序的运行结果相同。

然后构造了一个由英文字母构成的字符串“I have a great dream.”。用 strrchr()函数在字符串中查找“e”，那么函数返回字符串中最后一次出现“e”的位置之后的内容，程序中 3 次出现“e”，最后一次出现是在“dream”中，于是函数返回“eam.”（e 本身也被返回）。然后我们想试验一下是否可以把一个字符串作为参数传递给 strrchr()函数，于是在字符串中查找字符串“at”。如果该函数支持字符串参数，按照上面的分析，应当返回“at dream.”。但是看图 6-9 的运行结果，返回的却是“am.”。为什么呢？因为 strrchr()函数不支持字符串参数。如果提供了字符串参数，会自动截取字符串的第一个字符作为参数。也就是说参数“at”和参数“a”所起的作用一样。于是函数返回字符串中最后一次出现“a”之后的内容，也就是“am.”。

可能有读者会问，为什么要构造一个英文的字符串来讲解 strrchr()函数呢？通过刚才的分析就已经能够得到答案。因为每一个汉字都占两个字节，在函数中两个字节会被认为是多个字符（英文中一个字符占一个字节），因此，strrchr()函数就无法支持中文，也就是说不能把一个或多个中文字符作为参数传递给 strrchr()函数。

提 示

> 除 strrchr()函数之外，PHP 中还有很多函数无法直接处理中文，我们无法一一指出，读者在学习 PHP 和编写程序时应当多加注意。

6.3.5　字符串替换函数

字符串替换是 Web 编程当中极为常用的操作，比如要过滤掉用户提交的不文明的词语，

或者处理掉字符串中包含的危险脚本，替换掉某些关键词等等。PHP 提供了一些函数来完成字符串替换操作，如 nl2br()、str_replace()等。

1. nl2br()函数

该函数的名字看起来比较怪，中间包含一个数字“2”，用汉语念起来似乎有点别扭。实际上这里的“2”在英文中念“two”，与“to”谐音。这里的“2”实际上就是“to”的一种缩写而已。明白了这一点之后，函数名字和功能都一目了然了。在很多中文参考资料中，将此函数的功能描述为“将换行符替换成 HTML 的换行符
”，在本书中我们也沿用这一解释。但是如果查阅英文版 PHP 手册，会发现大意为“在每一行前插入 HTML 换行标记
”，也就是说是“插入”而不是“替换”。但是在使用此函数时，就其效果而言相当于“替换”，因此我们仍然采用一贯的解释，将其归为字符串替换函数。

下面通过一个简单例子来说明此函数的作用：

程序清单　文档 6-11.php

```
<!--文件 6-11：nl2br()函数的使用-->
<HTML>
<HEAD>
<TITLE>nl2br()函数的使用</TITLE>
</HEAD>
<BDOY>
<form action="6-11.php" method="post">
请输入一段包含回车的文字：<br>
<textarea cols="30" rows="6" name="content"></textarea>
<input type=submit value="提交看效果">
</form>
<?php
$content=$_POST["content"];
//如果用户输入内容不为空
if($content!=""){
 echo "<hr>";
 echo "直接输出接收到的内容：<br>";
 echo $content;
 echo "<br>（内容长度：".strlen($content)."）<br>";
 echo "<hr>";
 echo "用 nl2br()处理接收到的内容，然后输出：<br>";
 echo nl2br($content);
 echo "<br>（内容长度：".strlen(nl2br($content))."）<br>";
}
?>
</BDOY>
</HTML>
```

本程序首先创建了一个 TextArea 多行文本输入框，并要求输入一段包含回车的文字。之所以要求包含回车，就是因为 nl2br()函数处理的对象就是回车，我们无法测试其效果。我们不妨输入“子丑寅卯↙辰巳午未↙申酉戌亥”，其中“↙”表示按下键盘上的回车键。如图 6-10 所示。



图 6-10 nl2br()函数的使用（1）

单击“提交看效果”按钮，程序运行结果如图 6-11 所示。

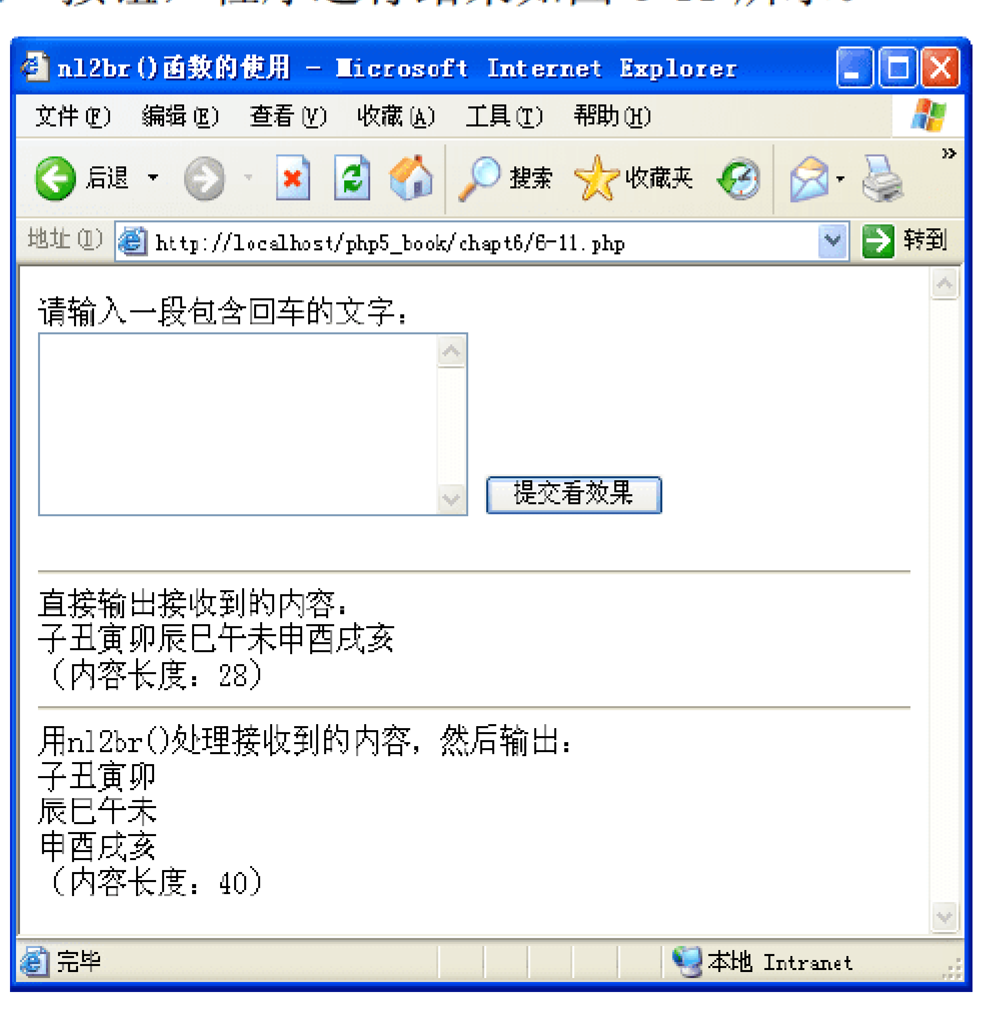


图 6-11 nl2br()函数的使用（2）

通过图 6-11 可以清楚地看到 nl2br()函数的效果。在未使用 nl2br()函数对接收到的内容进行处理时，本来输入三行内容，在网页中显示时全都连成了一行。这是因为 HTML 语言不识别回车换行符号。这一点在第 2 章学习 HTML 语言时已经有所体现，不论在 HTML 代码中连续打入多少个回车换行，都不会在网页上看到效果，就是因为浏览器会忽略掉 HTML 代码中的回车换行。当我们用 nl2br()对内容进行处理后，每一行前面都自动添加了一个“
”标记。这个标记就是我们通常用的 HTML 中的换行标记“
”，不过是写法略有不同而已。我们原本输入的三行内容，也便正常地显示出来。

此外，通过比较 nl2br()处理前和处理后的字符串长度，也可以看出此函数的工作原理。未处理之前提交的数据内容由 12 个汉字和两个回车换行构成，长度为 12*2+2*2=28（每次按下键盘的回车键都会产生一个换行和一个回车两个字符）。而用 nl2br()函数处理之后，数据内容长度变成 40，增加了 12 字节。而 12 字节恰好是 2 个“
”的长度（注意 br 和/之间的空格也占一个字节）。因此，足以证明 nl2br()函数的作用是在被处理的字符串中每一行之前插入一个“
”标记。它并没有替换掉任何内容。

虽然 nl2br()函数的本质并没有进行替换，但在使用中，其效果等同于将回车换行符号替换为 HTML 换行标记。因此在不严格要求的前提下，可以称之为字符替换函数。

2. str_replace()函数

PHP 提供的 str_replace()函数允许将一个字符串中的任意子串全部替换为另外一个子串。其使用格式如下：

```
mixed str_replace(mixed search,mixed replace,mixed subject [,int &count])
```

这个格式的简化解释如下：str_replace()函数将 subject 中的所有 search 替换成 replace，并把替换的次数存放在 count 中，其中 count 参数为可选。这里的 search、repalce、subject 以及整个函数的返回值都是 mixed 类型，也就是说提供的参数可以是多种类型，常用的有字符串和数组。下面来看一个例子：

程序清单　文档 6-12.php

```
<!--文件 6-12.php：字符串替换函数的使用-->
1:  <HTML>
2:  <HEAD>
3:   <TITLE>字符串替换函数综合范例</TITLE>
4:  </HEAD>
5:  <BDOY>
6:   <?php
7:    //单个字符替换
8:    $str="当所有的人[逗]离开我的时候[逗]你劝我要耐心等候[句]";
9:    echo "原字符串：<b>".$str."</b><br>";
10:   $str=str_replace("[","(",$str);
11:   $str=str_replace("]",")",$str);
12:   echo "字符替换之后：<b>".$str."</b><br>";
13:   //字符串替换
14:   $str=str_replace("(逗)","，",$str);
15:   $str=str_replace("(句)","。",$str);
16:   echo "字符串替换之后：<b>".$str."</b><br>";
17:  ?>
18: </BDOY>
19: </HTML>
```

在本例中，构造了一个字符串，其中逗号用“[逗]”表示，句号用“[句]”表示。然后第

10~11 行分别进行了两次替换，将字符“[”、“]”分别替换成“(”、“)”，然后输出替换后的字符串。在第 14~15 行，又进行了两次替换，将“(逗)”替换成“，”，将“(句)”替换成“。”，然后将最终的字符串输出。本程序运行结果如图 6-12 所示。

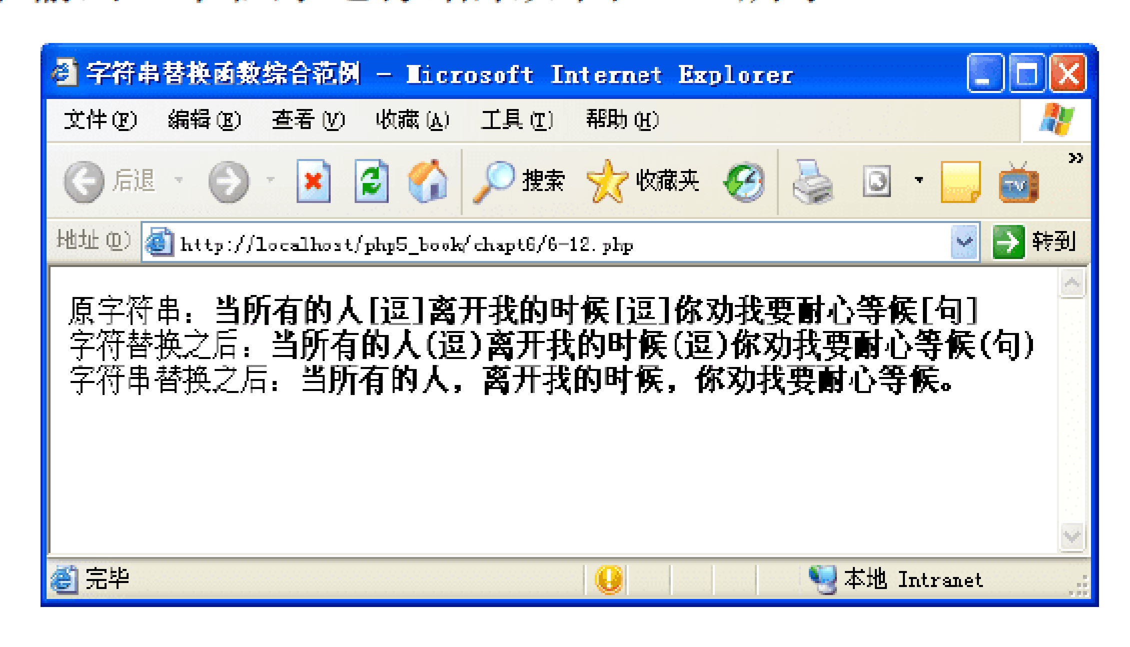


图 6-12　字符串替换函数使用范例

本例中主要用到了 str_replace()函数的普通字符串替换功能。str_replace()函数还可以接收数组参数，来实现批量的替换。将 6-12.php 作如下修改：

程序清单　文档 6-13.php

```
<!--文件 6-13：字符串替换的高级应用-->
<HTML>
<HEAD>
 <TITLE>字符串替换函数高级应用</TITLE>
</HEAD>
<BDOY>
 <?php
  //单个字符替换
  $str="当所有的人[逗]离开我的时候[逗]你劝我要耐心等候[句]";
  echo "原字符串：<b>".$str."</b><br>";
  $arr1= array("[","]");
  $arr2=array("(",")");
  $str=str_replace($arr1,$arr2,$str);
  echo "字符替换之后：<b>".$str."</b><br>";
  //字符串替换
  $arr3=array("(逗)","(句)");
  $arr4=array("，","。");
  $str=str_replace($arr3,$arr4,$str);
  echo "字符串替换之后：<b>".$str."</b><br>";
 ?>
</BDOY>
</HTML>
```

读者可以发现，修改之后的 6-13.php，在使用 str_replace()函数时传递了 2 个数组作为参数，第一个数组按顺序存放了要被替换的字符串，第二个数组按顺序存放了要替换成的字符

串。这样，不论要替换多少个字符串，只要按照顺序分别存放在 2 个数组中，然后调用一次 str_replace()函数即可完成。这样做有明显的优点，在要替换的项目很多的情况下，可以很大程度的简化程序。

本程序的运行结果与 6-12.php 完全相同，运行结果可参见图 6-12。

6.3.6 字符串截取函数

在编程中经常遇到要将一个字符串的一部分单独取出的情况，也就是字符串的截取。PHP 中常用字符串截取函数有 substr()等。

substr()函数的使用格式如下：

```
string substr(string string,int start [,int length])
```

本函数返回一个字符串中从指定位置开始指定长度的子串。参数 string 为原始字符串，start 为截取的起始位置（从 0 开始计），可选参数 length 为要截取的长度。值得一提的是，参数 start 和 length 均可以用负数，start 为负数时说明从倒数第 start 个字符开始取，length 为负数时表示从 start 位置开始取到倒数 length 的位置结束。下面看一个例子：

程序清单　文档 6-14.php

```
<!--文件 6-14.php：字符串截取综合范例-->
<HTML>
<HEAD>
 <TITLE>字符串的截取综合范例</TITLE>
</HEAD>
<BDOY>
 <?php
  //构造字符串
  $str="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  echo "原字符串：<b>".$str."</b><br>";
  //按各种方式进行截取
  $str1=substr($str,5);
  echo "从第 5 个字符开始取至最后：".$str1."<br>";
  $str2=substr($str,9,4);
  echo "从第 9 个字符开始取 4 个字符：".$str2."<br>";
  $str3=substr($str,-5);
  echo "取倒数 5 个字符：".$str3."<br>";
  $str4=substr($str,-8,4);
  echo "从倒数第 8 个字符开始向后取 4 个字符：".$str4."<br>";
  $str5=substr($str,-8,-2);
  echo "从倒数第 8 个字符开始取到倒数第 2 个字符为止：".$str5."<br>";
 ?>
</BDOY>
</HTML>
```

本程序运行结果如图 6-13 所示。

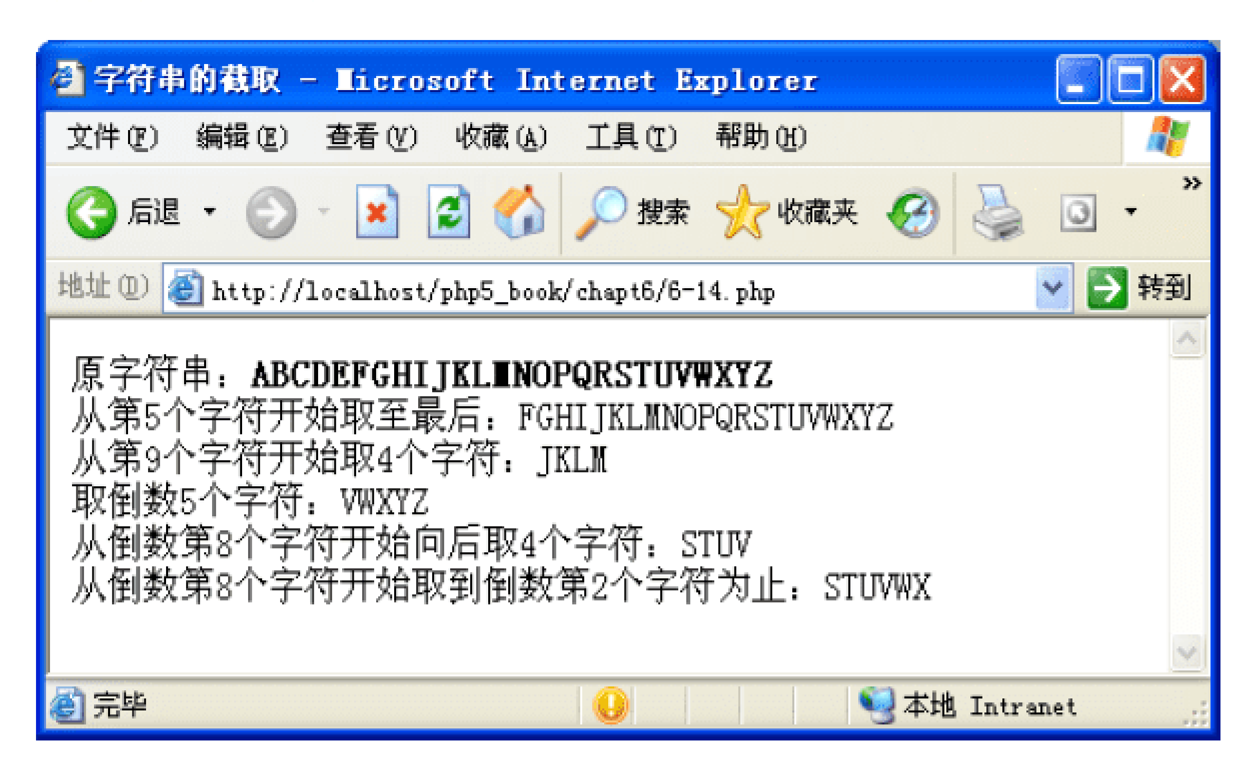


图 6-13　substr()字符串截取函数使用范例

读者可以对照图 6-13 来分析一下整个程序，应当对 substr()函数有一个深入的了解，尤其是 start 和 length 两个参数的含义和使用方法，更应该熟练掌握。有一点提醒读者注意：start 参数为正数时，从 0 开始计数。start 参数为负数时，从 1 开始计数。也就是说没有“倒数第 0 个字符”。读者可以参考本例加深理解，也可以自己动手编制一个程序来验证。

6.3.7　字符串分割函数

在编程中有时候需要将一个字符串按某种规则分割成多个。PHP 提供了 explode()、str_split()等函数来完成分割操作。下面分别介绍这 2 个函数。

1. explode()函数

explode()函数格式如下：

```
array explode(string separator,string string [,int limit])
```

explode()函数用来将一个字符串按照某个指定的字符分割成多段，并将各段按顺序存入一个数组中。该函数的返回值就是一个数组。separator 参数为分割符，可以是一个字符串，也可以是单个字符。string 为要处理的字符串。参数 limit 为可选，如果设置了 limit，则返回的数组包含最多 limit 个元素，并且最后那个元素将包含 string 的剩余部分。下面来看一个例子：

程序清单　文档 6-15.php

```
<!--文件 6-15.php：用 explode 函数分割字符串-->
<HTML>
<HEAD>
 <TITLE>explode 字符串分割函数</TITLE>
</HEAD>
<BDOY>
 <?php
```

```
 //构造字符串
 $str="苹果，空心菜，香蕉，萝卜，大蒜，牛肉";
 echo "原字符串：<b>".$str."</b><br>";
 echo "1.以逗号为分割符分割字符串：<br>";
 $arr1=explode("，",$str);
 echo "---\$arr1[0]的值：".$arr1[0]."<br>";
 echo "---\$arr1[4]的值：".$arr1[4]."<br>";
 echo "2.分割时指定 limit 参数：<br>";
 $arr2=explode("，",$str,3);
 echo "---\$arr2[0]的值：".$arr2[0]."<br>";
 echo "---\$arr2[2]的值：".$arr2[2]."<br>";
 echo "---\$arr2[4]的值：".$arr2[4]."<br>";
?>
 </BDOY>
</HTML>
```

在本例中，定义了一个普通字符串$str。字符串中出现了多个逗号，用 explode()函数来分隔这个字符串，把“，”作为分割字符。在未提供 limit 参数的情况下，字符串被成 6 小段，并存入一个数组$arr1 中。每一小段分别对应$arr[0]，$arr[1]…$arr[5]。然后指定 limit 参数为 3，再次用 explode()函数分隔字符串$str，这时返回的数组$arr2 只包含 3 个元素，即$arr2[0]，$arr2[1]，$arr2[2]。这时$arr2[2]中存放的不是第三个逗号之前的内容，而是前两个逗号之后的所有内容。

本程序运行结果如图 6-14 所示。

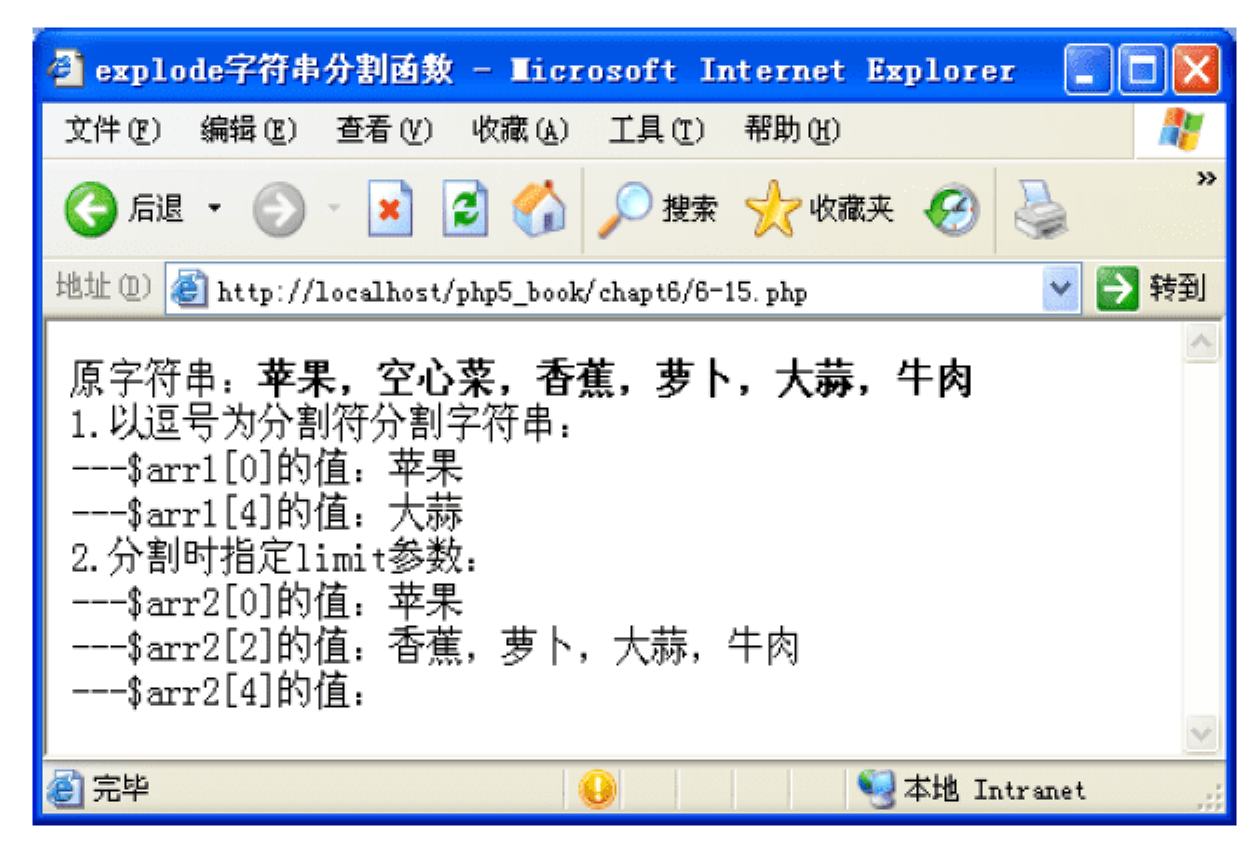


图 6-14　explode 函数分隔字符串使用范例

2. str_split()函数

str_split()函数格式如下：

```
array str_split(string string [,int split_length])
```

str_split()函数将一个字符串以一定长度为单位分割成多段，并返回由各段组成的数组。str_split()函数不是以某个字符串为分割依据，而是以一定长度为分割依据。参数 string 为

要分割的字符串，可选参数 length 设置分割的单位长度。具体例子如下：

程序清单　文档 6-16.php

```
<!--文件 6-16.php：用 str_split 函数分割字符串-->
<HTML>
<HEAD>
 <TITLE>str_split 字符串分割函数</TITLE>
</HEAD>
<BDOY>
 <?php
  //分割英文字符串
  $str="Quietly I leave,just as quietly I came.";
  echo "原字符串：<b>".$str."</b><br>";
  echo "1.以默认长度分割字符串：<br>";
  $arr1=str_split($str);
  echo "---\$arr1[0]的值：".$arr1[0]."<br>";
  echo "---\$arr1[1]的值：".$arr1[1]."<br>";
  echo "---\$arr1[10]的值：".$arr1[10]."<br>";
  echo "2.以指定长度 5 分割字符串：<br>";
  $arr2=str_split($str,5);
  echo "---\$arr2[0]的值：".$arr2[0]."<br>";
  echo "---\$arr2[1]的值：".$arr2[1]."<br>";
  echo "---\$arr2[5]的值：".$arr2[5]."<br>";
  //测试分割中文
  $str2="轻轻地我走了，正如我轻轻地来。";
  echo "原字符串：<b>".$str2."</b><br>";
  echo "1.以指定长度 5 分割字符串：<br>";
  $arr3=str_split($str2,5);
  echo "---\$arr3[0]的值：".$arr3[0]." <br>";
  echo "---\$arr3[1]的值：".$arr3[1]." <br>";
  echo "2.以指定长度 4 分割字符串：<br>";
  $arr4=str_split($str2,4);
  echo "---\$arr4[0]的值：".$arr4[0]." <br>";
  echo "---\$arr4[1]的值：".$arr4[1]." <br>";
  echo "---\$arr4[4]的值：".$arr4[4]." <br>";
  ?>
</BDOY>
</HTML>
```

本程序中，首先构造了一个英文字符串。然后用 str_split()函数直接分割。分割之后字符串被一个字符一个字符的分割开来，并且将这些字符顺次存放到数组$arr1 中。通过图 6-15的输出结果，能够印证这一点。接下来指定分割的单位为 5，这时候字符串按 5 个字符一段被分割成多段，并存储在数组$arr2 中。这时同样可以看到正确的输出结果。

前面已经提到多次，一个汉字字符占 2 个字节。很多字符串处理函数对中文支持的并不好。为了测试 str_split()函数分割中文的效果，我们又构造了一个字符串，这个字符串全部由汉字字符构成。首先用 5 作为分割单位来分割字符串，通过输出结果可以看出，str_split 函数无法区别中文，如第 1 段，取出的不是“轻轻地我走”5 个汉字字符，而是“轻轻?”。这里为什么会有一个“?”呢？这个问号不是字符串中的，而是在分割字符串时将第 3 个字“地”分割成了两段，因此无法正确显示这个字符，只能显示为“?”。

为了解决这个问题，又用一个偶数 4 作为分割长度，这时候汉字可以正确显示，整个字符串以 2 个汉字字符为单位被分割成多段。也就是说，在分割中文时，分割长度必须是 2 的倍数，否则将会导致汉字被分成两段而无法正确显示。

纵然如此，分割方案还有不完美之处，那就是当一个字符串是由中英文或者中文与阿拉伯数字混合而成，那即使是用 2 的倍数作为分割长度，仍然无法避免汉字被分割的情况。如字符串“110 是一个重要的电话号码”，如果以 2、4 作为分割长度，都会导致“是”这个汉字被分割。因此在使用 str_split()函数时必须充分考虑汉字的影响，否则会产生不可预料的结果。本程序运行结果如图 6-15 所示。

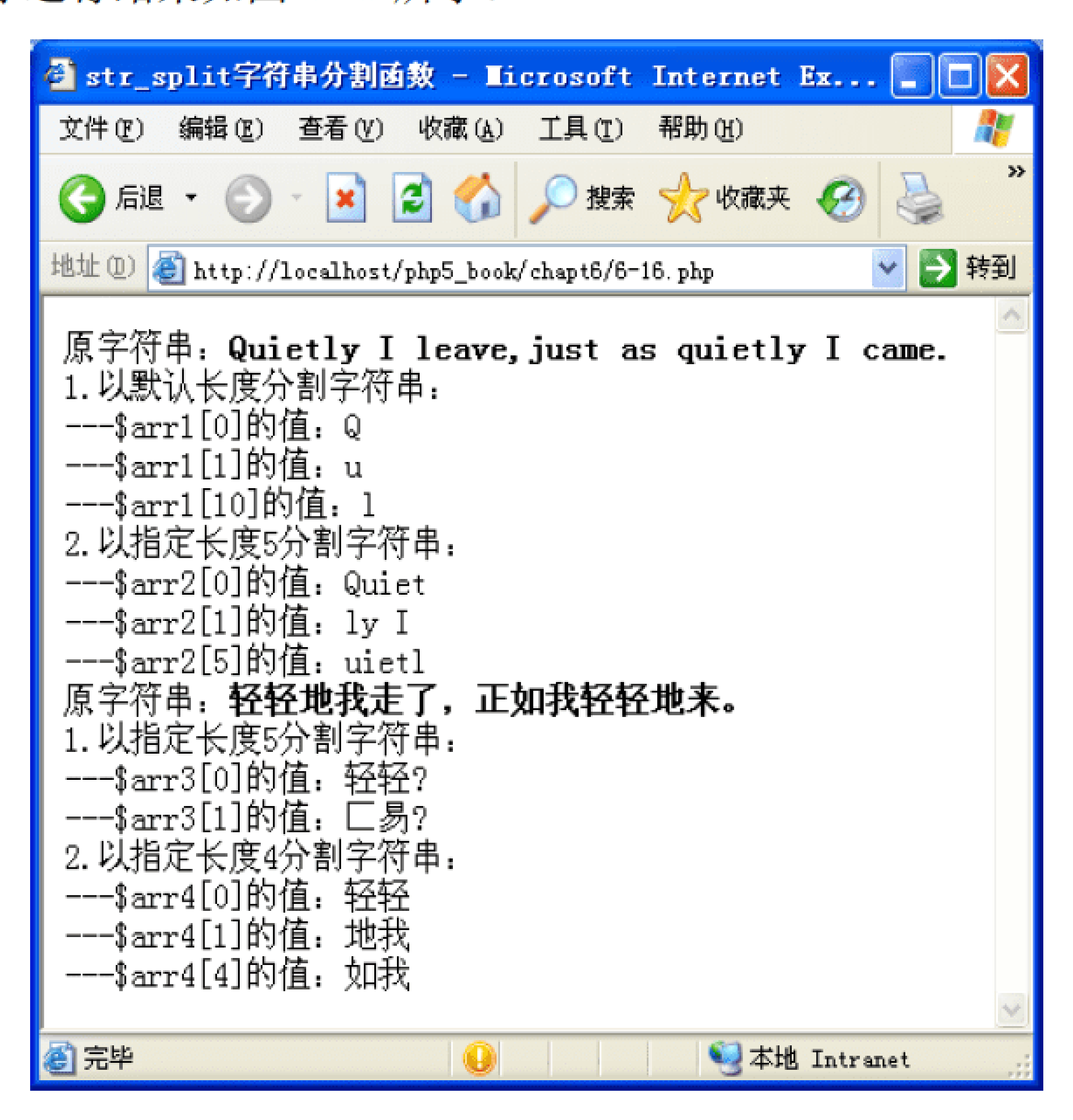


图 6-15　str_spilt()函数分割字符串使用范例

关于字符串处理函数，就介绍到这里。字符串处理函数在编程中使用极为频繁，读者应当熟练掌握，多多积累。本节中介绍的都是字符串处理函数中最为常用的部分，另外还有大量颇有价值的函数，限于篇幅无法一一介绍，读者可以参考表 6-1 及 PHP 官方手册自行学习、掌握，为后面学习深入的 PHP 编程打下基础。

另外，虽然在讲解时每个函数都是独立地来讲，但是读者应注意这些函数的结合使用。在一个程序中，可能会同时用到多个函数，通过多个函数的综合应用来实现一个操作，读者应在这方面多下功夫。

6.4　时间日期函数

6.4.1　时间日期函数一览

时间日期函数用来取得服务器的时间和日期，以及对时间日期类型的数据进行各种处理，来满足程序的需要。在编程中时常要用到时间日期，比如信息发布时要记录发布的时间，用户注册时要记录注册的时间，以及记录用户进行某些操作的时间等等。PHP 5 提供的时间日期函数如表 6-3 所示。

表 6-3　PHP 时间日期函数

函数名	函数功能概述
checkdate	验证一个格里高里日期
date_default_timezone_get	取得一个脚本中所有日期时间函数所使用的默认时区
date_default_timezone_set	设定用于一个脚本中所有日期时间函数的默认时区
date_sunrise	返回给定的日期与地点的日出时间
date_sunset	返回给定的日期与地点的日落时间
date	格式化一个本地时间 / 日期
getdate	取得日期 / 时间信息
gettimeofday	取得当前时间
gmdate	格式化一个 GMT/UTC 日期 / 时间
gmmktime	取得 GMT 日期的 UNIX 时间戳
gmstrftime	根据区域设置格式化 GMT/UTC 时间 / 日期
idate	将本地时间日期格式化为整数
localtime	取得本地时间
microtime	返回当前 Unix 时间戳和微秒数
mktime	取得一个日期的 Unix 时间戳
strftime	根据区域设置格式化本地时间 / 日期
strptime	解析由 strftime()生成的日期 / 时间
strtotime	将任何英文文本的日期时间描述解析为 Unix 时间戳
time	返回当前的 Unix 时间戳

通过表 6-3 可以看到，PHP 开发者提供了数十个函数来实现各种时间/日期操作。其中不乏很有趣的函数，如返回某给定日期与地点的日出/日落时间，不过这其中部分函数并没有很大的实用价值。我们只需要熟练掌握其中几个函数的用法，即可实现绝大多数函数常见的应用，下面选最常用的进行介绍。

6.4.2　获取当前时间的 Unix 时间戳

很多读者可能不明白什么叫做 Unix 时间戳。Unix 时间戳是指从 Unix 纪元（格林威治时间 1970 年 1 月 1 日 00 时 00 分 00 秒）开始到当前时间为止相隔的秒数。很显然 Unix 时

间戳应该代表一个很大的整数。Unix 时间戳在很多时候非常有用，尤其在对时间进行加减时作用最为明显。比如当前时间是“2006 年 10 月 10 日 10 点 10 分 10 秒”，在这个时间基础上加上 25 天 8 小时 55 分 58 秒，会得到一个什么时间呢？推算起来可能比较复杂，因为除了时间进位以外，还涉及到不同月份天数可能不同（可能是 28 天、29 天、30 天、31 天）等等，所以用数学方法直接加减是不行的。如果我们使用 Unix 时间戳，那么在第一个时间的基础上加上一定的秒数，得到的就是第二个时间的 Unix 时间戳。然后用 PHP 的有关函数把这个时间戳转换成普通时间格式显示即可。

PHP 中提供了 time()函数来直接获取当前时间的 Unix 时间戳。看下面的代码：

程序清单 文档 6-17.php

```
<--文件 6-17.php：获取当前时间的 Unix 时间戳-->
<HTML>
<HEAD>
<TITLE>获取 Unix 时间戳</TITLE>
</HEAD>
<BDOY>
<?php
$tm= time();
echo "当前时间的 Unix 时间戳为：".$tm;
?>
</BDOY>
</HTML>
```

本程序运行结果如图 6-16 所示。

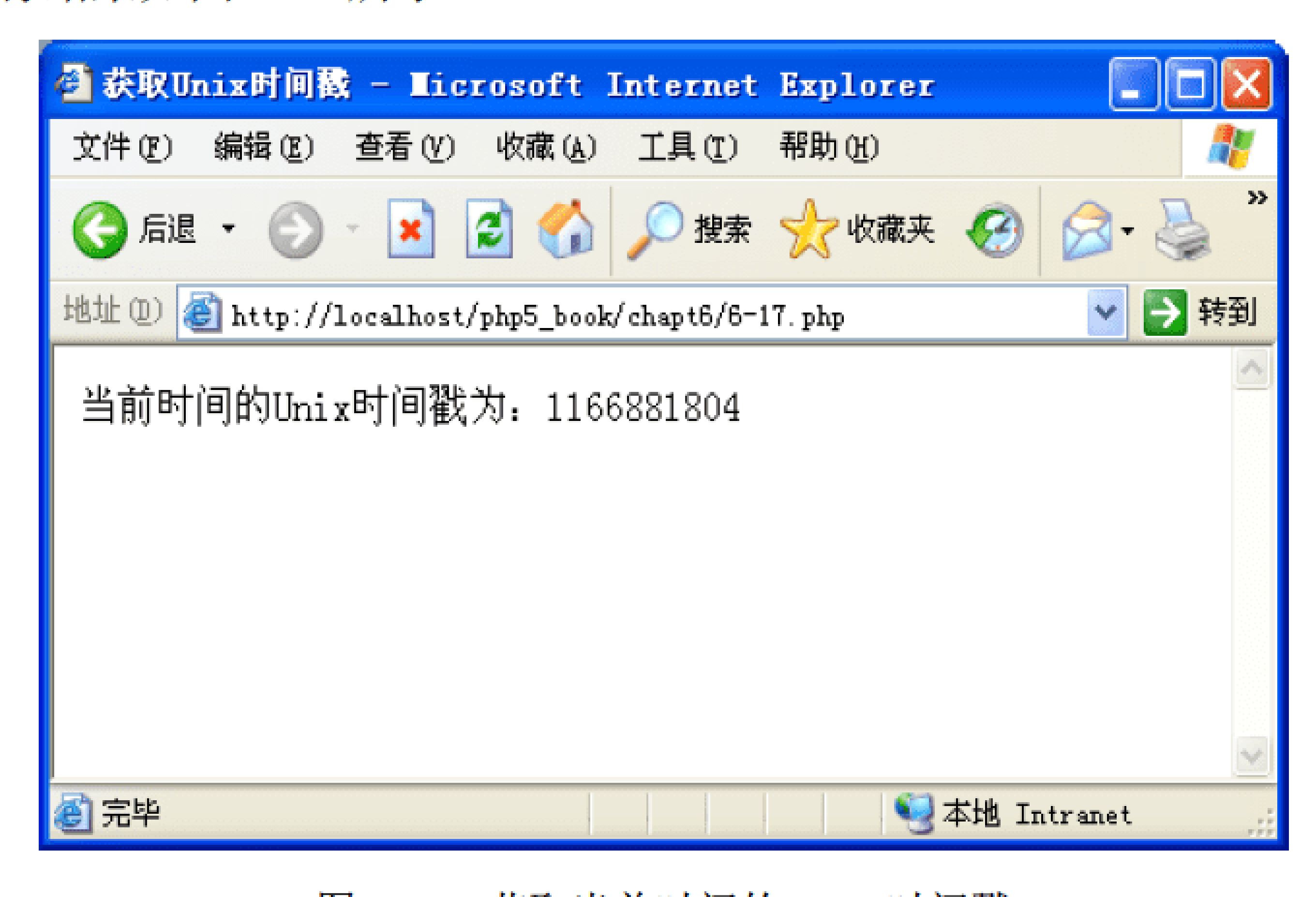


图 6-16 获取当前时间的 Unix 时间戳

图中的数字 1166881804 表示的是从 1970 年 1 月 1 日 0 点 0 分 0 秒到本程序执行时相隔的秒数。如果每隔一段时间刷新一下页面，会发现时间戳的值每次都会变化，因为每过一秒钟，这个数字就增加 1。每次刷新页面都会重新调用程序，每次都会获得一个不同的时间，因此 Unix 时间戳在不断变化。具体的说，这个数字在不断增大。

6.4.3　获取指定时间的 Unix 时间戳

能不能获得一个指定时间的Unix时间戳呢？也就是说要获得的时间戳不是当前时间的，而是一个固定时间的。PHP 提供了 mktime()函数和 strtotime()函数来完成这个操作。其使用格式如下：

```
int mktime([int hour [,int minute [,int second [,int month [,int day [,int
year]]]]]])
```

本函数的作用是根据给出的参数返回 Unix 时间戳。6 个参数全部都是整数，分别代表小时、分钟、秒、月、日、年。参数可以从右向左省略，省略的参数会被设置成本地日期和时间的当前值。当全部参数都被省略时，获得的就是当前时间的 Unix 时间戳。

另外 strtotime()函数允许我们使用一个时间字符串作为参数来获取 Unix 时间戳。这个时间串的顺序与中文习惯较为吻合。如“2000-11-12 10:34:55”表示 2000 年 11 月 12 日 10 时 34 分 55 秒。该字符串指代了一个具体的时间，可以作为 strtotime()函数的参数，来获得这个时间的 Unix 时间戳。我们看一个例子：

程序清单　文档 6-18.php

```
<--文件 6-18.php：获取指定时间的 Unix 时间戳-->
<HTML>
<HEAD>
<TITLE>获取指定时间的 Unix 时间戳</TITLE>
</HEAD>
<BDOY>
<?php
//用 mktime()返回时间戳
 $tm= mktime(23,56,59,12,20,1999);
 echo "1999 年 12 月 20 日 23 点 56 分 59 秒的 Unix 时间戳为：".$tm;
 //用 strtotime()返回时间戳
 $tm2= strtotime("1999-12-20 23:56:59");
 echo "<br>用 strtotime 获得的同一时间的时间戳：".$tm2;
?>
</BDOY>
</HTML>
```

本程序运行结果如图 6-17 所示。

本程序中用 mktime()函数将一个指定时间格式化为 Unix 时间戳。然后用 strtotime()函数格式化这个时间，不过是用一个字符串而不是 6 个数字作为参数。运行结果显示两个函数的返回结果完全相同。不论你如何刷新这个页面，时间戳值都不会发生变化。因为这个时间戳是一个固定的时间，不会随当前时间变化而变化。

有的读者可能会问一个问题：如果我使用 mktime()函数时提供的参数不符合常规，会出现什么情况呢？比如每年最多有 12 个月，不可能有 14 月，如果我给月份这个参数提供

一个 14，会怎样呢？实际上，PHP 会把某年的 14 月作为下一年的 2 月。如 1999 年 14 月会被认为是 2000 年 2 月。同理如果时间设置为 23 分 70 秒，那 PHP 会作为 24 分 10 秒来处理。读者可以自行编制程序试验一下。

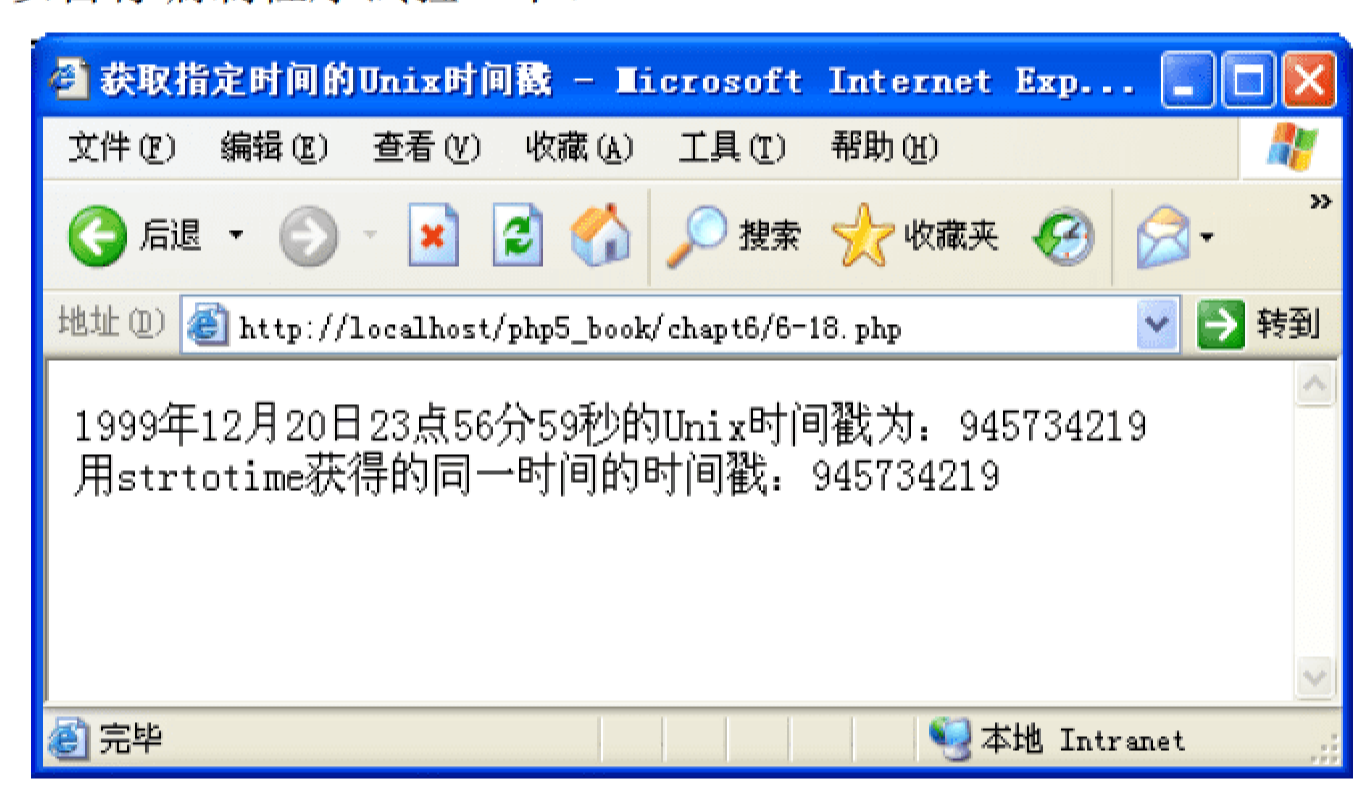


图 6-17 获取指定时间的 Unix 时间戳

6.4.4 从 Unix 时间戳取得时间日期信息

前面学习了如何获得一个时间的 Unix 时间戳。虽然用 Unix 时间戳有利于在计算机中进行时间的计算，但是在显示时间时还是应该显示成通用的“年月日时分秒”以及星期几等格式，而不是直接输出一个 Unix 时间戳。PHP 中提供了 date()和 getdate()等函数来实现从 Unix 时间戳到通用时间日期的转换。

1. getdate()函数

getdate()函数用来将一个 Unix 时间戳格式化成具体的时间日期信息，其使用格式如下：

```
array getdate([int timestamp])
```

其中参数 timestamp 就是一个 Unix 时间戳。如果不指定参数，则默认使用当前时间。该函数返回一个数组，数组中存放了详细的时间信息。通过数组下标可以取得数组中的元素值。其下标与值得对应关系如表 6-4 所示。

表 6-4 getdate()函数返回数组的下标与值对应表

下标	说明	返回值例子
"seconds"	秒的数字表示	0 到 59
"minutes"	分钟的数字表示	0 到 59
"hours"	小时的数字表示	0 到 23
"mday"	月份中第几天的数字表示	1 到 31
"wday"	星期中第几天的数字表示	0（表示星期天）到 6（表示星期六）
"mon"	月份的数字表示	1 到 12
"year"	4 位数字表示的完整年份	例如：1999 或 2003

（续表）

下标	说明	返回值例子
"yday"	一年中第几天的数字表示	0 到 365
"weekday"	星期几的完整文本表示	Sunday 到 Saturday
"month"	月份的完整文本表示	January 到 December
0	该时间的 Unix 时间戳	整数

下面看一个例子，来全面展示该函数的强大功能。

程序清单　文档 6-19.php

```
<!--文件 6-19.php：获取详细的时间信息-->
1: <HTML>
2: <HEAD>
3:  <TITLE>getdate()函数获取详细的时间信息</TITLE>
4: </HEAD>
5: <BDOY>
6:  <?php
7:   //首先假设一个时间
8:   $dt= "2007-07-01 08:30:00";
9:   echo "时间：".$dt."<br>";
10:  //将此时间格式化为 Unix 时间戳
11:  $tm= strtotime($dt);
12:  echo "此时间的 Unix 时间戳：".$tm."<br>";
13:
14:  //获取该时间的详细信息
15:  $arr=getdate($tm);
16:  //输出详细信息
17:  echo "秒：".$arr["seconds"]."<br>";
18:  echo "分：".$arr["minutes"]."<br>";
19:  echo "时：".$arr["hours"]."<br>";
20:  echo "日：".$arr["mday"]."<br>";
21:  echo "月：".$arr["mon"]."/".$arr["month"]."<br>";
22:  echo "年：".$arr["year"]."<br>";
23:  echo "星期：".$arr["wday"]."/".$arr["weekday"]."<br>";
24:  echo "该日期是该年中的第".$arr["yday"]."天<br>";
25: ?>
26:</BDOY>
27:</HTML>
```

本程序中，在第 8 行设置了一个时间，第 10 行将此时间格式化成 Unix 时间戳。第 15 行将此时间戳用 getdate()函数获取详细时间信息，然后在第 17~24 行分别输出了全部的时间信息。程序的输出结果如图 6-18 所示。

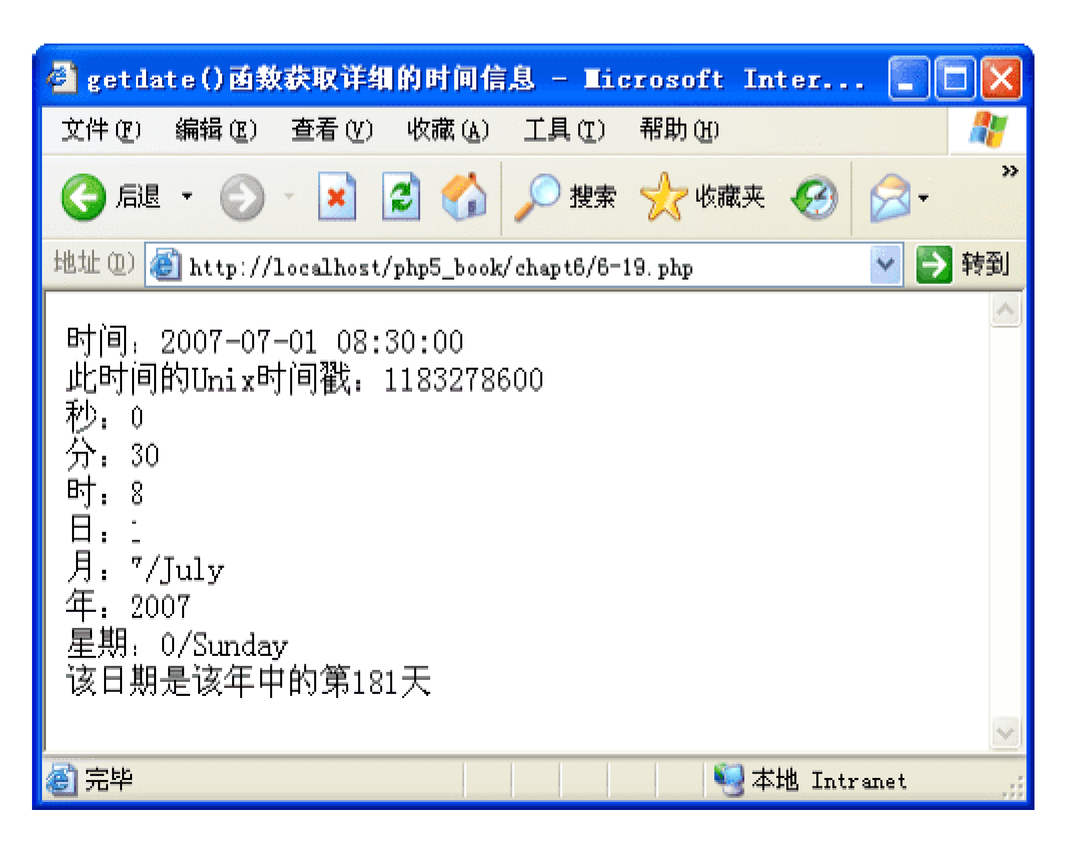


图 6-18 用 getdate()函数获取详细的时间信息

本程序中假定日期为“2007-07-01 08:30:00”，实际上可以直接用语句“$arr=getdate();”来获得当前时间的详细信息。这时输出的时间信息就是当前程序执行时的时间信息。感兴趣的读者可以自行试验一下。

2. date()函数

date()函数用来将一个 Unix 时间戳格式化成指定的时间/日期格式。上面讲的 getdate()函数可以获取详细的时间信息，但是很多时候并不要取得如此具体的时间信息，而是将一个 Unix 时间戳所代表的时间按照某种容易识读的格式输出来，这就需要用到 date()函数。该函数的使用格式是：

```
string date(string format [,int timestamp])
```

该函数直接返回一个字符串。这个字符串就是一个指定格式的日期时间。参数 format 是一个字符串，用来指定输出的时间的格式。可选参数 timestamp 是要处理的时间的 Unix 时间戳。如果参数为空，那么默认值为当前时间的 Unix 时间戳。

本函数的重点是学习如何使用 format 参数。format 参数必须由指定的字符构成，不同的字符代表不同的特殊含义（见表 6-5）。

表 6-5 format 字符一览表

format 字符	说明	返回值例子
d	月份中的第几天，有前导零的 2 位数字	01 到 31
D	星期中的第几天，文本表示，3 个字母	Mon 到 Sun
j	月份中的第几天，没有前导零	1 到 31
l（小写的“L”）	星期几，完整的文本格式	Sunday 到 Saturday
N	ISO-8601 格式数字表示的星期中的第几天（PHP 5.1.0 新加）	1（表示星期一）到 7（表示星期天）
w	星期中的第几天，数字表示	0（表示星期天）到 6（表示星期六）
z	年份中的第几天	0 到 366

（续表）

format 字符	说明	返回值例子
W	ISO-8601 格式年份中的第几周，每周从星期一开始（PHP 4.1.0 新加的）	例如：42（当年的第 42 周）
F	月份，完整的文本格式，例如 January	January 到 December
m	数字表示的月份，有前导零	01 到 12
M	三个字母缩写表示的月份	Jan 到 Dec
n	数字表示的月份，没有前导零	1 到 12
t	给定月份所应有的天数	28 到 31
L	是否为闰年	如果是闰年为 1，否则为 0
Y	4 位数字完整表示的年份	例如：1999 或 2003
y	2 位数字表示的年份	例如：99 或 03
a	小写的上午和下午值	am 或 pm
A	大写的上午和下午值	AM 或 PM
B	Swatch Internet 标准时	000 到 999
g	小时，12 小时格式，没有前导零	1 到 12
G	小时，24 小时格式，没有前导零	0 到 23
h	小时，12 小时格式，有前导零	01 到 12
H	小时，24 小时格式，有前导零	00 到 23
i	有前导零的分钟数	00 到 59
s	秒数，有前导零	00 到 59
e	时区标识（PHP 5.1.0 新加）	例如：UTC，GMT，Atlantic/Azores
I	是否为夏令时	如果是夏令时为 1，否则为 0
O	与格林威治时间相差的小时数	例如：+0200
T	本机所在的时区	例如：EST，MDT 等

表 6-5 中列出了绝大部分 format 字符，个别极为不常用的没有列出。通过表 6-5 已经看出 format 字符数量众多，涉及方方面面。date()函数可以完成的功能极为丰富。

下面通过一个例子来看这些 format 字符如何使用。

程序清单　文档 6-20.php

```
<!--文件 6-20.php：用格式化字符串输出时间日期-->
1:<HTML>
2:<HEAD>
3: <TITLE>用格式化字符串输出时间日期</TITLE>
4:</HEAD>
5:<BDOY>
```

```
<?php
 //设置一个时间（如采用当前时间可用 time()）
 $tm=strtotime("2006-09-09 10:30:40");
 echo "初始化设置的时间为：2006-09-09 10:30:40<br>";
 //使用不同的格式化字符串测试输出效果
  echo date("Y-M-D H:I:S A",$tm)."<br>";
  echo date("y-m-d h:i:s a",$tm)."<br>";
  echo date("Y 年 m 月 d 日[l] H 点 i 分 s 秒",$tm)."<br>";
  echo date("F,d,Y l",$tm)."<br>";
  echo date("Y-M-D H:I:S",$tm)."<br>";
  echo date("Y-M-D H:I:S",$tm)."<br>";
  echo date("所在时区：T，与格林威治时间相差：O 小时",$tm)."<br>";
 ?>
</BDOY>
</HTML>
```

本程序的运行结果如图 6-19 所示。

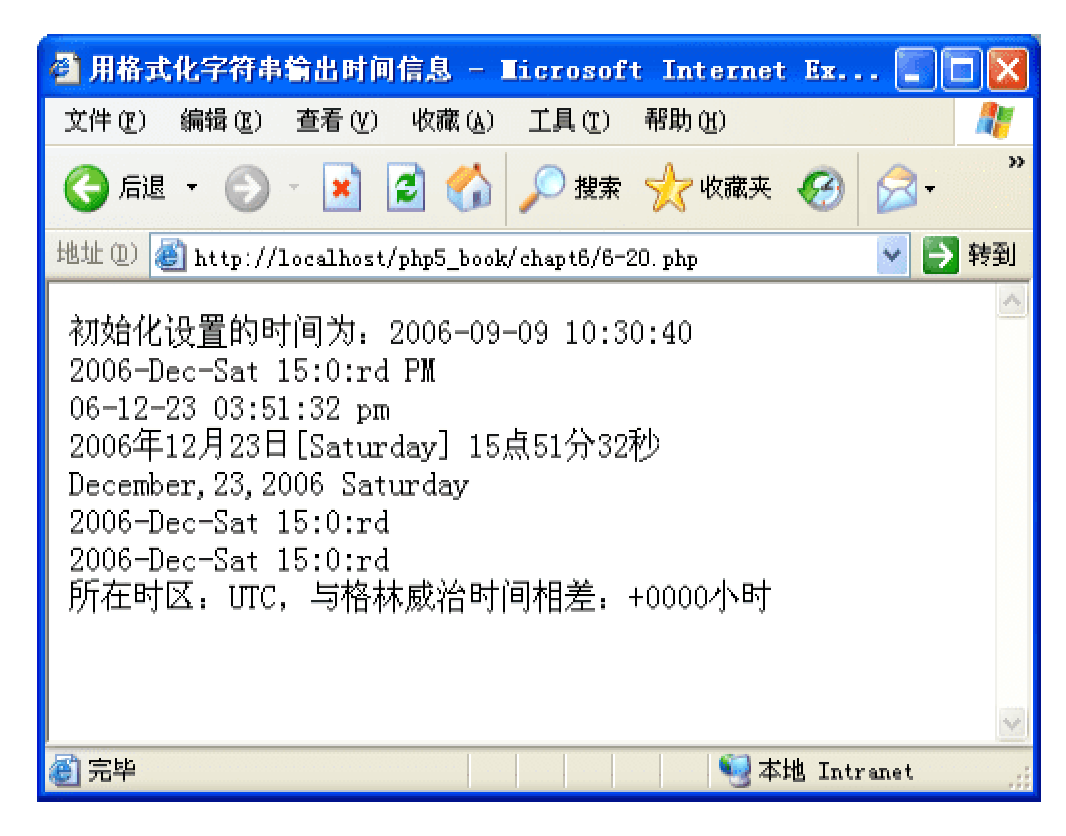


图 6-19　用格式化字符串输出时间日期

通过程序 6-20.php 可以看出，格式化字符串的使用非常灵活。只要在字符串中包含相关字符，date()函数就能把这些字符替换成指定的时间日期信息。我们可以利用这个函数随意输出自己需要的时间日期格式。

程序的最后一条输出用的是格式字符“T”和“O”来输出运行本程序的服务器所处的时区以及本时区和格林威治标准时间相差的小时数。程序输出时区为 UTC，相差时间为 0 小时。这虽然与世界标准时区和时间相符，但是并不是本地的时间。比如中国要比格林威治时间晚 8 个小时，因此在取得的本地时间基础上再增加 8 个小时才是北京时间。增加 8 个小时的方法很简单，在已经取得的当前时间的 Unix 时间戳上加 8*60*60 即是 8 小时之后的时间戳。如果读者在编写程序时发现程序获得的时间与北京时间不符，应该考虑是时区问题，对取得的时间进行相应处理即可。

PHP 的时间日期函数很常用，但并不复杂。读者一般只需要掌握 Unix 时间戳的获得和操作方法，掌握格式化字符的使用方法，即可轻松掌握 PHP 时间日期的处理。

6.5　数 学 函 数

编程中少不了要进行数据计算操作。除了基本的加、减、乘、除等运算以外，还有求正弦值、余弦值、绝对值、对数值、取整、取余、进制转换以及生成随机数等一系列操作。这些操作都可以通过简单的函数调用来实现。这就是 PHP 的数学函数。

表 6-6 列出了常用的数学函数及其功能。

表 6-6　常用数学函数一览

函数名	功能
abs	绝对值
acos	反余弦
acosh	反双曲余弦
asin	反正弦
asinh	反双曲正弦
atan2	两个参数的反正切
atan	反正切
atanh	反双曲正切
base_convert	在任意进制之间转换数字
bindec	二进制转换为十进制
ceil	进一法取整
cos	余弦
cosh	双曲余弦
decbin	十进制转换为二进制
dechex	十进制转换为十六进制
decoct	十进制转换为八进制
deg2rad	将角度转换为弧度
exp	计算 e（自然对数的底）的指数
floor	舍去法取整
fmod	返回除法的浮点数余数
getrandmax	显示随机数最大的可能值
hexdec	十六进制转换为十进制
hypot	计算一直角三角形的斜边长度
is_finite	判断是否为有限值
is_infinite	判断是否为无限值

（续表）

函数名	功能
is_nan	判断是否为合法数值
lcg_value	组合线性同余发生器
log10	以 10 为底的对数
log	自然对数
max	找出最大值
min	找出最小值
mt_getrandmax	显示随机数的最大可能值
mt_rand	生成更好的随机数
mt_srand	播下一个更好的随机数发生器种子
octdec	八进制转换为十进制
pi	得到圆周率值
pow	指数表达式
rad2deg	将弧度数转换为相应的角度数
rand	产生一个随机整数
round	对浮点数进行四舍五入
sin	正弦
sinh	双曲正弦
sqrt	平方根
srand	播下随机数发生器种子
tan	正切
tanh	双曲正切

本类函数虽然数目众多，但是使用方法较为简单，函数功能对照表 6-6 可一目了然。下面构造一个例子，来说明几个常用数学函数的使用方法。

首先产生一个大于等于 100 小于等于 200 的随机数，把这个数字作为一个圆的面积，然后根据这个面积算出圆的半径，并将得到的半径数字用不同的方法取整。程序如下：

程序清单　文档 6-21.php

```
<!--文件 6-21.php：数学函数使用举例-->
<HTML>
<HEAD>
 <TITLE>数学函数使用举例</TITLE>
</HEAD>
<BDOY>
 <?php
```

```
$s=rand(100,200);
$pi=pi();
$r=sqrt($s/$pi);
$qz1=round($r);   //四舍五入取整
$qz2=ceil($r);     //进一法取整
$qz3=floor($r);    //舍去法取整

echo "随机产生的圆的面积为：".$s."<br>";
echo "通过除法和开方计算出的圆的半径为：".$r."<br>";
echo "四舍五入取整后：".$qz1."<br>";
echo "进一法取整后：".$qz2."<br>";
echo "舍去法取整后：".$qz3."<br>";
?>
</BDOY>
</HTML>
```

在本程序中，用到了 6 个数学函数，分别用来产生一定范围内的随机数、返回圆半径（精确到小数点后 14 位）、开方以及取整等。程序本身比较简单，我们不再详细讲解，本程序运行结果如图 6-20 所示。

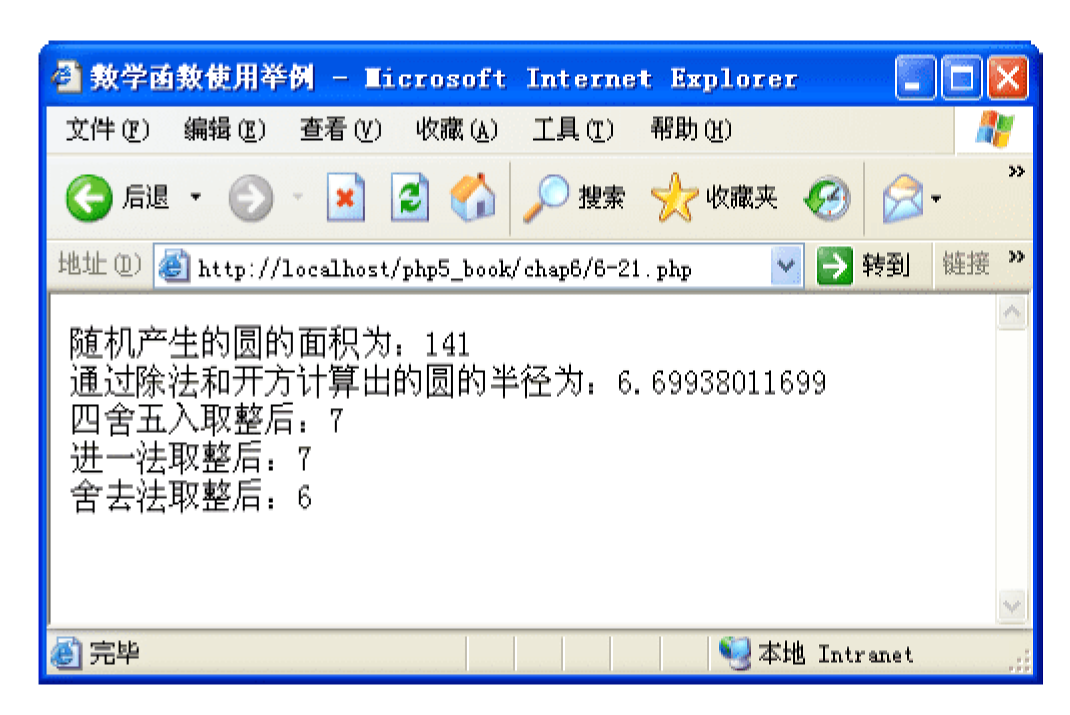


图 6-20 用常用数学函数计算圆的半径

由于圆的面积是随机产生的，所以每次刷新这个页面，都会重新产生一组数据。读者可以自行试验，通过不同的数据来分析程序中函数的作用。

数学函数虽然数目众多，但是使用都较为简单，本书限于篇幅，不再一一介绍。请读者参考 PHP 手册，尝试其他函数的使用方法。

6.6 图像处理函数

PHP 提供了一系列函数，来实现在网站编程中对图像进行编辑。虽然这些函数能够实现的功能十分有限，无法和功能强大的专业图形图像软件相比，但是在需要动态生成图像、自动批量处理图像等方面，能给 PHP 网站开发者带来巨大帮助。其中最为典型的应用有随机图形验证码、图片水印、数据统计中饼状图、柱状图的生成等等。

PHP 的图像处理函数都封装在一个函数库中，这就是 GD 库。要使用 GD 库中的函数

来进行图像处理，必须保证安装了 GD 库。在 PHP 官方的标准发行版本中，都包含了这个库。如本书中介绍的 PHP 5 版本，这个 GD 库存放在 PHP 安装目录下的 ext 子目录下，名为 php_gd2.dll。读者如果担心自己的 PHP 版本是否包含这个函数库，可以打开 PHP 安装目录查找一下。

并不是 php_gd2.dll 库文件存在，就可以使用这些函数了。在默认的 php.ini 设置中，这个库并不自动载入。所以，需要首先打开这个库的自动载入功能，这样这个库中的函数就像 PHP 标准函数一样可以直接在程序中使用了。打开的方法很简单，用记事本打开 php.ini 配置文件，利用查找功能找到“;extension=php_gd2.dll”这一行，将最前面的分号去掉，然后保存，重新启动 IIS（Apache），这时候 GD 库已经被自动加载了。如图 6-21 所示。

```
;extension=php_curl.dll
;extension=php_dba.dll
;extension=php_dbase.dll
;extension=php_exif.dll
;extension=php_fdf.dll
;extension=php_filepro.dll
extension=php_gd2.dll
;extension=php_gettext.dll
;extension=php_ifx.dll
;extension=php_imap.dll
;extension=php_interbase.dll
;extension=php_ldap.dll
```

图 6-21　开启 GD 图像函数库

若要确定此函数库是否已经被成功加载，可以打开第 3 章中讲过的 PHP 环境信息显示程序，也就是 phpinfo()程序，查看列出的信息中是否有 GD 一项。该项目中详细列出了当前 PHP 的 GD 库信息。如图 6-22 所示。

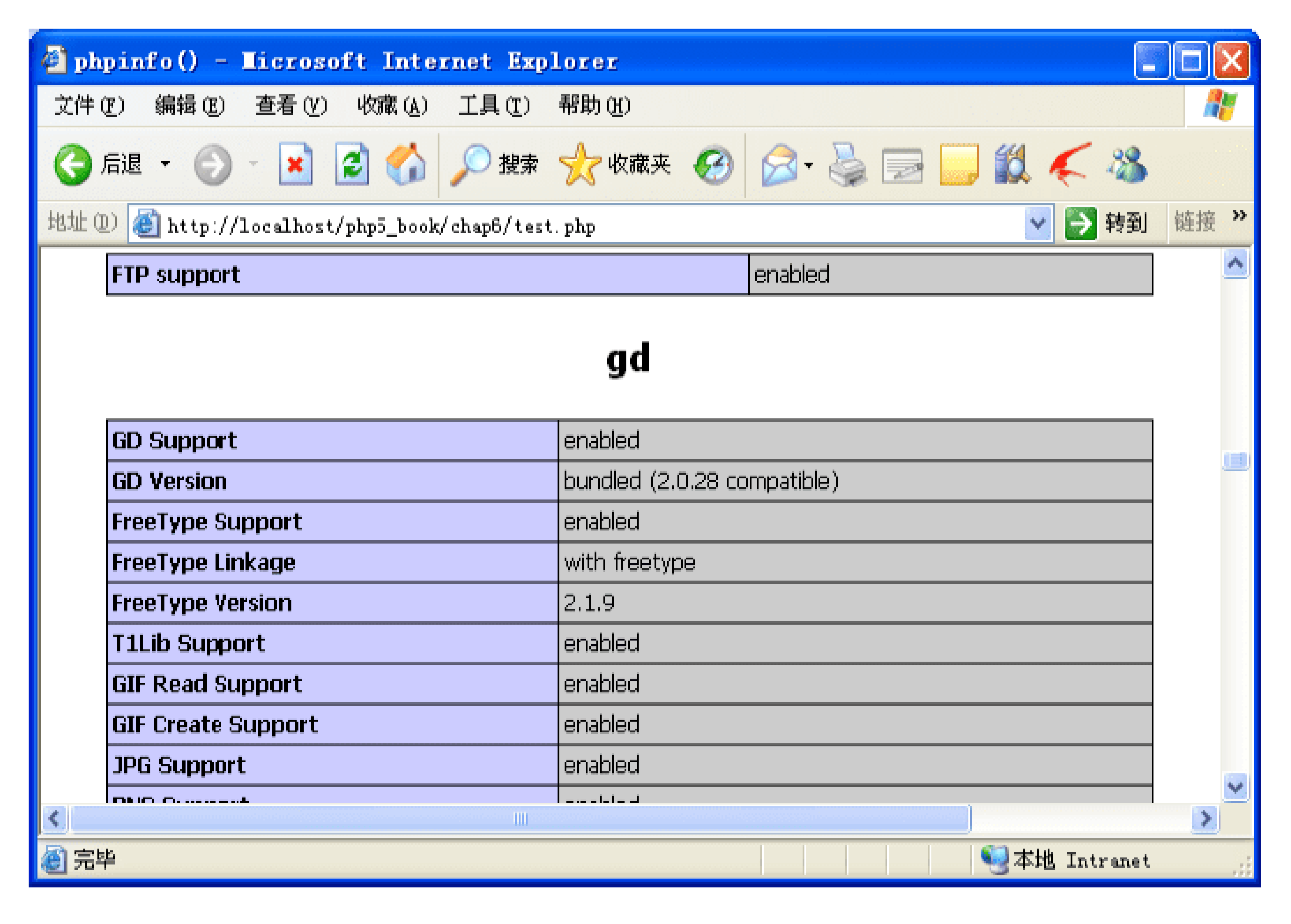


图 6-22　GD 库信息

现在已经做完了使用 GD 库之前的所有准备工作，下面就来看 GD 库中究竟有哪些函数，如下表 6-7 所示。

表 6-7　PHP 常用图像处理函数一览

函数名	功能
gd_info	取得当前安装的 GD 库的信息
getimagesize	取得图像大小
image_type_to_extension	取得图像类型的文件后缀
imagearc	画椭圆弧
imagechar	水平地画一个字符
imagecharup	垂直地画一个字符
imagecopy	拷贝图像的一部分
imagecopymerge	拷贝并合并图像的一部分
imagecopyresized	拷贝部分图像并调整大小
imagecreatefromgd2	从 GD2 文件或 URL 新建一图像
imagecreatefromgd	从 GD 文件或 URL 新建一图像
imagecreatefromgif	从 GIF 文件或 URL 新建一图像
imagecreatefromjpeg	从 JPEG 文件或 URL 新建一图像
imagecreatefrompng	从 PNG 文件或 URL 新建一图像
imagecreatefromstring	从字符串中的图像流新建一图像
imagecreatetruecolor	新建一个真彩色图像
imagedashedline	画一虚线
imagedestroy	销毁一图像
imageellipse	画一个椭圆
imagefill	区域填充
imagefilledarc	画一椭圆弧且填充
imagefilledellipse	画一椭圆并填充
imagefilledpolygon	画一多边形并填充
imagefilledrectangle	画一矩形并填充
imagegd2	将 GD2 图像输出到浏览器或文件
imagegd	将 GD 图像输出到浏览器或文件
imagegif	以 GIF 格式将图像输出到浏览器或文件
imageistruecolor	检查图像是否为真彩色图像
imagejpeg	以 JPEG 格式将图像输出到浏览器或文件
imageline	画一条线段
imageloadfont	载入一新字体

（续表）

函数名	功能
imagepalettecopy	将调色板从一幅图像拷贝到另一幅
imagepng	以 PNG 格式将图像输出到浏览器或文件
imagepolygon	画一个多边形
imagepstext	用 PostScript Type1 字体把字符串画在图像上
imagerectangle	画一个矩形
imagerotate	用给定角度旋转图像
imagesetbrush	设定画线用的画笔图像
imagesetpixel	画一个单一像素
imagesetstyle	设定画线的风格
imagesetthickness	设定画线的宽度
imagesettile	设定用于填充的贴图
imagestring	水平地画一行字符串
imagestringup	垂直地画一行字符串
imagesx	取得图像宽度
imagesy	取得图像高度

PHP 5 提供的图像处理函数总数超过了 100 个，表 6-7 中仅列出了部分常用函数。

6.6.1 用图像处理函数绘制 Png 图形

由于本部分函数个数较多，而且具体使用方法较为复杂，要在很有限的篇幅内进行一个较全面的介绍是十分困难的。这里仅给出两个例子，可以先体会一下 PHP 图像处理函数的简单使用。表中列出的其他函数以及未在表中列出的函数，可以参考 PHP 手册进行深入学习。此外在本章的末尾有一个完整的图形验证码的实例来进一步介绍图像处理函数。

程序清单　文档 6-22.php

```
<!--文件 6-22.php：图像处理函数使用举例-->
<?php
header("Content-type:image/png");
$im=@imagecreate(200,100)or die("无法创建图像流");
@imagecolorallocate($im,240,150,255);
$t_color1=imagecolorallocate($im,0,0,0);
$t_color2=imagecolorallocate($im,100,100,100);
imagestring($im,5,8,10,"I like PHP!",$t_color1);
imagestringup($im,5,8,90,"Hello!",$t_color2);
imageellipse($im,65,65,55,55,$t_color1);
imagefilledrectangle($im,110,95,160,30,$t_color2);
```

```
11:imagepng($im);
12:imagedestroy($im);
13:?>
```

本程序创建了一幅 png 图像，并且在该图像上进行绘图操作。程序运行结果如图 6-23 所示。

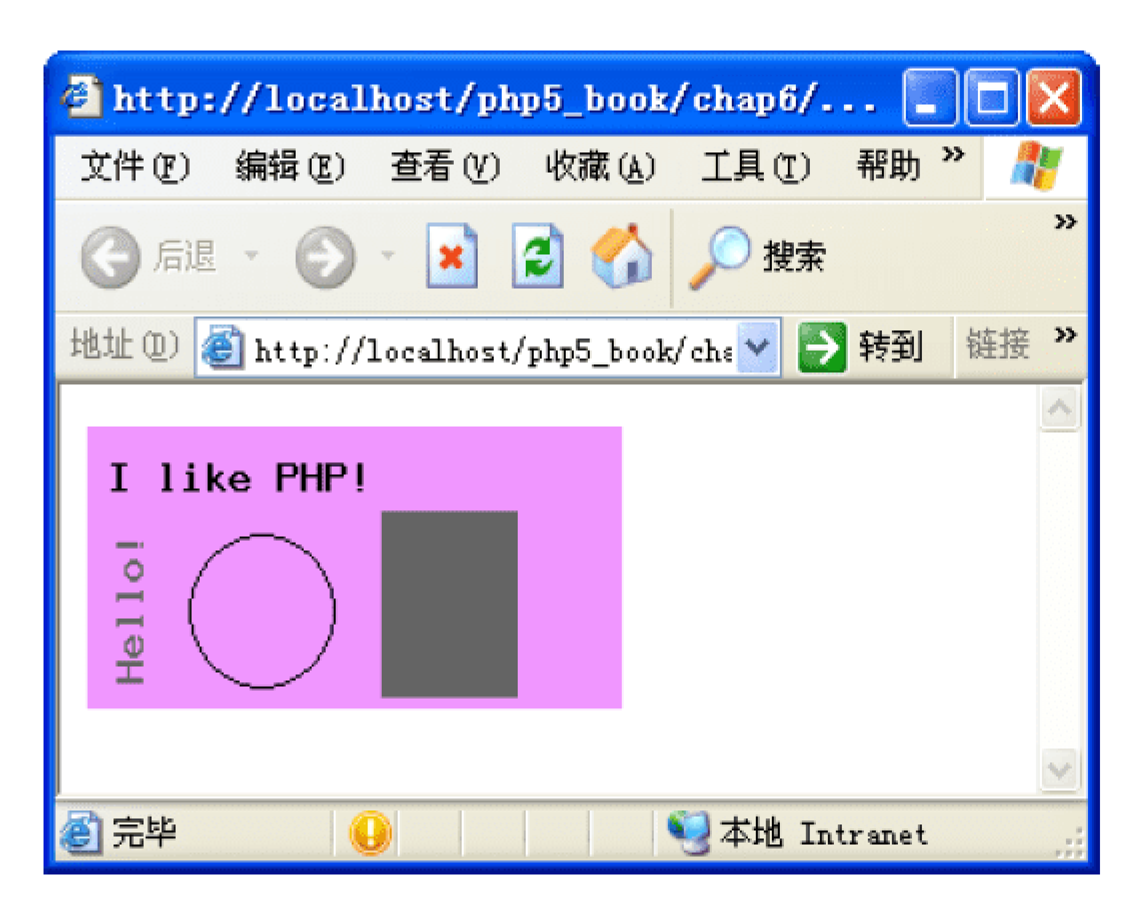


图 6-23　用 PHP 直接绘图

本程序第 2 行指定了网页的类型，即 png 图像。这样 6-22.php 虽然是一个 PHP 程序，但是其作用是动态生成一张图像，几乎等同于一张图像。在本程序中，普通的输出语句如 echo 等都是无效的，这一点应当注意。

第 3 行用 imagecreate()函数创建一幅新图像，两个参数为图像的宽度和高度，单位是像素，并且返回此图像的数据流，存放于$im 变量中。

第 4 行用 imagecolorallocate()函数设置了图像的背景颜色。4 个参数分别为图像流、R 色值、G 色值、B 色值。3 个色值合并即产生了 RGB 色值。这里的 240，150，255 运行之后显示淡紫色。另外（0，0，0）为黑色，（255，255，255）为白色，（255，0，0）为红色等等。关于 RGB 颜色的有关详细信息请自行查阅有关资料。

第 5~6 行分别生成了 2 种颜色，存在不同的变量中以备后面使用。第一种为黑色，第二种为浅灰色。

第 7 行用 imagestring()函数在图像上“写入”了一个字符串。6 个参数分别表示图像流、所用字体、写入点的 x 坐标、写入点的 y 坐标、要写入的字符串、字符串颜色。这里面有两点值得注意：一是函数的第 2 个参数取值范围为 1~5，分别代表了不同大小和是否加粗的 5 种字体，可以试着修改此参数来观察程序运行效果；二是这里的 x,y 坐标都是相对于图像的左上角，最左上角坐标为（0，0），向右为 x 轴，向下为 y 轴，单位都是像素。

第 8 行用 imagestringup()函数来向图像中竖向写入一个字符串。函数的参数含义与 imagestring()函数相同。

第 9 行用 imageellipse()函数在图像中绘制了一个圆。函数第一个参数为图像流，最后一个为绘图所用颜色，第 2、3、4、5 个参数分表表示圆心的 x 坐标、圆心的 y 坐标、圆的 x 方向半径长度、圆的 y 方向半径长度。在本例中，绘制了一个圆心在（65,65），半径为

55 的圆。如果要绘制一个椭圆，只需要确定圆心位置，然后分别设置 x 方向半径和 y 轴方向半径即可。当这两个半径相等时是一个圆，不相等时是一个椭圆。

第 10 行用 imagefilledrectangle()函数绘制了一个矩形，并对矩形进行颜色填充。第一个参数为图像流，最后一个参数为填充颜色，第 2、3、4、5 个参数的含义分别为矩形左上角 x 坐标、矩形左上角 y 坐标、矩形右下角 x 坐标、矩形右下角 y 坐标。也就是说只要提供矩形的左上角和右下角坐标，即可绘制此矩形。

第 11 行用 imagepng()函数将此图像流输出为一张 png 格式的图片，也就是在浏览器中显示的图片。

第 12 行销毁了这个图像流。

6.6.2 用图像处理函数制作水印效果

除了 png 格式，还可以把图像输出为 jpg、gif 等常用的格式。而且不仅可以直接创建一个图像流来绘制图形，还可以将一张已有的图片作为图像流读入，然后在此基础上对图像进行处理，这一功能常用来制作图像水印等。所谓图像水印，就是在图像上标上一些特殊的图形或符号，用来作为图像所有者的标记或者防止图片被盗用。下面就看一个这样的例子。

在本例中，有一张原始图片 pic.jpg，现在用 PHP 将此图片进行处理，在图片表面按一定规律加上文字标记，产生水印效果，使之不能被直接盗用。

程序清单 文档 6-23.php

```
<!-- 文件 6-23.php：为图片加水印-->
1:<?php
2:header("Content-type:image/jpeg");
3:$im=imagecreatefromjpeg("pic.jpg");
4:$white=imagecolorallocate($im,255,255,255);
5:$width=imagesx($im);
6:$height=imagesy($im);
7:$x=0;
8:$y=0;
9:while($x<$width && $y<$height){
10:imagestring($im,2,$x,$y,"http://www.xxx.com",$white);
11:$x+=20;
12:$y+=20;
13:}
14:imagejpeg($im);
15:imagedestroy($im);
16:?>
```

本程序第 2 行设定本页面输出类型为 jpeg 图像。

第 3 行用 imagecreateformjpeg()函数打开了一张图片 pic.jpg，并返回此图片的数据流。

第 4 行定义了一个颜色（白色）。

第 5~6 行用 imagesx()和 imagesy()函数取得图片 pic.jpg 的原始尺寸。

第 7~8 行定义了用于控制文字添加位置的两个变量。

第 9~13 行用循环向图片中添加多行文字，用$x 和$y 两个变量控制位置和循环次数。

第 14 行输出此图片，第 15 行销毁数据流。

程序运行前和运行后图像如图 6-24、图 6-25 所示。

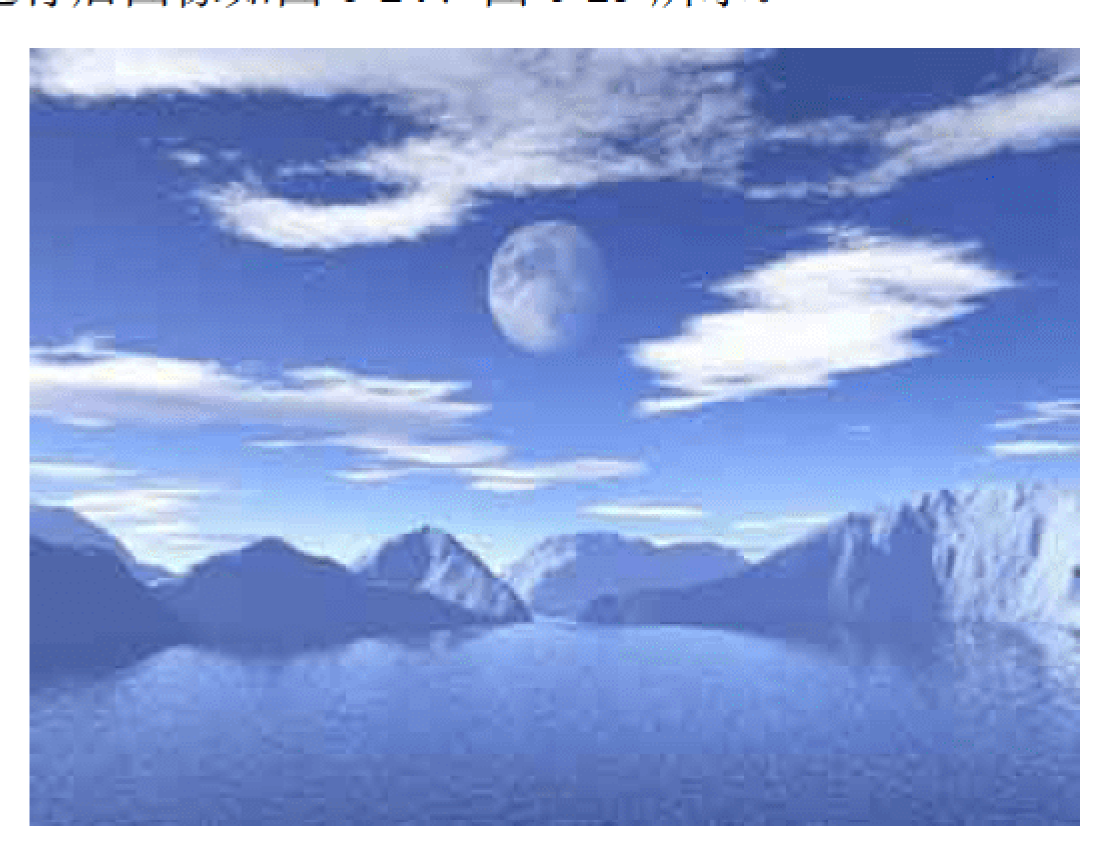


图 6-24　原始图片



图 6-25　加水印后的图片

可以看到，处理后的图片上被加入了文字标记，这就基本达到了处理目的。但是同时处理后的图片由于文字的加入影响了图像的观赏性。因此水印如何加，加在什么位置，既能起到水印的作用，又不影响美观，是在处理中最应考虑的。

PHP 的图像处理函数，就介绍这些。希望能够使读者对此有一个基本的了解，为以后深入学习打下基础。

6.7 其 他 函 数

据粗略统计，PHP 5 提供的函数总数达 3800 多个，分属于 160 多个类别，体系极为庞大。这些函数涵盖了 PHP 编程的方方面面，给 PHP 开发者带来巨大的便利和强有力的支撑。

虽然本章中已经用了很大的篇幅来介绍一些最常用的函数，但和PHP全部函数比起来，仍然只是极小的一部分。除此之外，还有数据库函数、XML 函数、Socket 函数、正则表达式函数、COM 与 DOM 函数、压缩函数、MAIL 函数等等。这么多的函数必然不是短期学习能够掌握的，这需要首先充分了解 PHP 函数的体系，然后通过长时间不断的学习、积累，最终达到较高的水平。

学习过程中，建议读者必备一份中文版的官方 PHP 手册，它在 PHP 官方网站和国内其他网站上很容易下载到。这本手册涵盖了 PHP 方方面面的知识，并提供了几乎全部函数的介绍。可以说 PHP 手册是 PHP 初学者以及 PHP 开发者必不可少的参考工具。

关于 PHP 函数部分，就介绍这些。本书中的其他章节还将根据需要介绍其他函数，如文件/目录函数、数据库函数等。

6.8 案例 6 完整图像验证码程序开发

6.8.1 案例分析

图像验证码程序是当前 Web 开发中十分常用的程序。本章中学习了 PHP 的图像处理函数，结合前面章节学习的 Session 函数以及表单数据提交技术等，可以写出一个完整的图像验证码程序。

验证码在网站中的作用一般是防止恶意“灌水”，也就是防止恶意发布垃圾信息。如果没有验证码，那么攻击者可以利用辅助软件实现自动提交、自动注册等，由于软件执行的效率高、速度快且可以连续工作不疲倦，因此常被用来攻击某个网站，制造大量垃圾数据，严重影响网站正常运行。

采用了验证码的方式之后，由于验证码每次都不一样，只有验证码输入正确才能提交信息，这样一来辅助软件就无法随意向服务器提交信息。因此，验证码的设计也有一些原则，如验证码的生成是随机的，无规律可循。另外，有的辅助软件有文字识别功能，能够从图片中辨析出文字，因此验证码中的数字可以采用随机的颜色，而且七扭八歪不易辨认。总之，最理想的验证码应该是用肉眼可以很容易地辨认出来，但是用软件识别就极为困难。

鉴于此，在设计这个验证码程序的时候，就不是简单的创建一幅图片，然后随机生成几个数字写上去，而是应再加入一些干扰。用 PHP 提供的图像处理函数，可以在图像上加入一些密密麻麻的像素点，然后随机绘制上两条虚线，再将几个数字的位置打乱。这样，机器识别就变得十分困难了。

本实例中用到以下三个文件：

- showimg.php：生成验证码，将验证码写入图片，并输出图片；
- login.html：调用 showimg.php，将用户输入的验证码提交到 check.php 进行验证；
- check.php：验证用户输入的验证码是否正确。

6.8.2 实施步骤

1. 生成验证码

按照上面的分析，结合学过的知识，编写代码如下：

程序清单 文档 showimg.php

```
<!-- 文件 showimg.php：生成验证码图片的程序-->
<?php
    //随机生成一个 4 位数的数字验证码
    $num="";
    for($i=0;$i<4;$i++){
    $num .= rand(0,9);
    }
    //4 位验证码也可以用 rand(1000,9999)直接生成
    //将生成的验证码写入 session，备验证页面使用
    Session_start();
    $_SESSION["Checknum"]=$num;
    //创建图片，定义颜色值
     Header("Content-type:image/PNG");
     $im=imagecreate(60,20);
     $black=ImageColorAllocate($im,0,0,0);
     $gray=ImageColorAllocate($im,200,200,200);
     imagefill($im,0,0,$gray);
     //随机绘制两条虚线，起干扰作用
   $style=array($black,$black,$black,$black,$black,$gray,$gray,$gray,
   $gray,$gray);
    imagesetstyle($im,$style);
    $y1=rand(0,20);
    $y2=rand(0,20);
    $y3=rand(0,20);
    $y4=rand(0,20);
    imageline($im,0,$y1,60,$y3,IMG_COLOR_STYLED);
    imageline($im,0,$y2,60,$y4,IMG_COLOR_STYLED);
    //在画布上随机生成大量黑点，起干扰作用；
    for($i=0;$i<80;$i++)
    {
    imagesetpixel($im,rand(0,60),rand(0,20),$black);
    }
    //将数字显示在画布上,数字之间的水平与垂直距离都在一定范围内随机生成
```

```
$strx=rand(3,8);
for($i=0;$i<4;$i++){
$strpos=rand(1,6);
imagestring($im,5,$strx,$strpos,substr($num,$i,1),$black);
$strx+=rand(8,12);
}
ImagePNG($im);
ImageDestroy($im);
?>
```

程序中的重要位置都已经做了注释，在此不再详细讲解。本程序运行后可以在浏览器中生成一张带有验证码的图片，每次刷新程序都会生成一个新验证码。以下是分别刷新三次后产生的验证码图片效果，如图 6-26 所示。

图 6-26　生成的验证码图片效果

仔细揣摩此程序，学习 PHP 图像操作中的直线、点的生成方法。

2. 调用验证码

编写一个普通网页，来调用验证码图片，并给出表单要求用户输入。

程序清单　文档 login.html

```
<!-- 文件 login.html：调用图片，生成表单-->
<HTML>
<HEAD>
<TITLE>图形验证码程序</TITLE>
</HEAD>
<BODY>
<form action="check.php" method="post">
<img src=showimg.php><br>
请输入验证码：<input type="text" name="passcode">
<input type=submit value="确定">
</form>
</BODY>
</HTML>
```

本程序是一段纯 HTML 代码，无需多做解释。惟一值得一提的是，在调用这个图片时，采用“<img src=showimg.php>”的方式。因为验证码图片本身是一张图片，所以使用<img>标记来引用，而这张图片又是用 PHP 程序生成的，因此直接用“src=showimg.php”来调用。

本网页运行结果如图 6-27 所示。

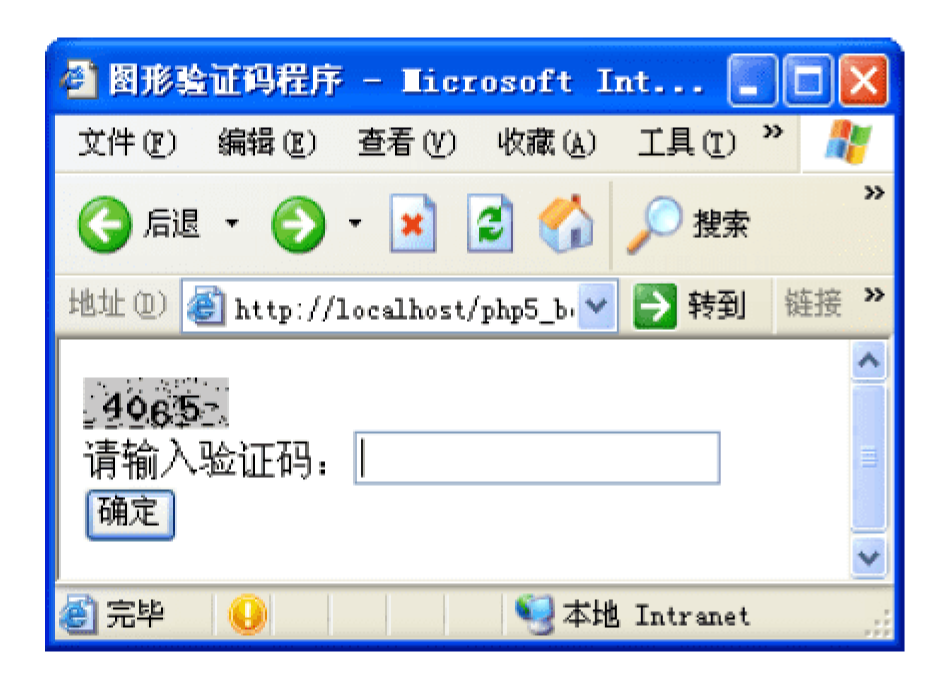


图 6-27 图形验证码调用页面

3. 验证程序

程序清单 文档 check.php

```
<!--文件 check.php：对输入的验证码进行验证-->
1:<?php
2:session_start();
3:$passcode=$_SESSION["Checknum"];
4:$usercode=$_POST["passcode"];
5:if($passcode == $usercode){
6: echo "验证码正确！验证通过！";
7: }else{
8: echo "验证码输入错误！验证失败！";
9: }
10:?>
```

程序第 3 行将 Session 中存储的正确的验证码读取出来，第 4 行将用户输入的验证码接收过来，然后进行比较，如果相等，说明用户输入的验证码正确，否则不正确。

到此为止，本实例就开发完成了。由于本实例重在展示图形验证码的创建和调用，没有加入过多元素，使得这个程序看起来非常简单，实用性不强。事实上这个程序作为一个独立的验证码程序，可以很容易地嵌入到任何用户注册、登录、信息发布等系统中，来实现图形验证码的功能。

6.9 本 章 小 结

本章中着重介绍了 PHP 5 的常用函数，并介绍了 PHP 函数的总体体系。主要讲解了数组函数、字符串处理函数、日期时间函数、图像处理函数等常用函数。学完本章后应熟练掌握 20~30 个常用 PHP 函数的使用方法，并能够参考 PHP 手册自行学习其他函数的使用方法。

6.10 练习题

1. 从网络上下载 PHP 手册，了解 PHP 的函数体系。
2. 编制程序，练习数组函数、字符串处理函数的使用。
3. 常用的时间/日期函数有哪些？在使用时应注意什么问题？
4. 编写一个绘图程序，在图像上绘制一个简易人物形状，并适当添加文字信息。

CHAPTER 7

PHP 5 文件与目录操作



本章重点

- 文件操作
- 目录操作
- 文件上传的实现

7.1 文件操作

在网络编程中要用到的文件操作大致可以分为两大类，一种是普通文件的操作，另一种是数据库文件的操作。在普通文件的操作中，对记事本文件的操作最为简单。下面就来探讨一下 PHP 对文件（以记事本为例）的操作。

实际上，PHP 对文件的操作已经规定好了若干个内置函数。要想利用 PHP 对文件进行操作，就是在 PHP 中对相关文件处理函数的应用。现在就常用的文件处理函数讲解如下。

1. resource fopen(string filename,string mode [,bool use_include_path])

该函数用于打开文件或者 URL。

其中，string filename 是要打开的文件名，必须为字符串形式。如果 filename 是"scheme://..."（如 http://...）的格式，则被当成一个 URL，PHP 将用搜索协议处理器（也被称为封装协议）来处理此模式。如果 PHP 认为 filename 指定的是一个本地文件（如"num.txt"），将尝试在该文件上打开一个流，该文件必须是 PHP 可以访问的，因此需要确认文件访问权限允许该访问。

string mode 是打开文件的方式，必须为字符串形式，其可能值列表如下：

- 'r'，只读形式，文件指针指向文件的开头。
- 'r+'，可读可写，文件指针指向文件的开头。
- 'w'，只写形式，文件指针指向文件的开头，打开同时清除所有内容，如果文件不存在，将尝试建立文件。
- 'w+'，可读可写，文件指针指向文件的开头，打开同时清除所有内容，如果文件不存在，将尝试建立文件。
- 'a'，追加形式（只可写入），文件指针指向文件的最后，如果文件不存在，将尝试建立文件。
- 'a+'，可读可写，文件指针指向文件的最后，如果文件不存在，将尝试建立文件。

2. string fgets(int handle [,int length])

该函数用于从文件指针中读取一行。

其中，int handle 是要读入数据的文件流指针，由 fopen()函数返回数值；int length 是要读入的字符个数，实际读入的字符个数是 length-1。

从 handle 指向的文件中读取一行并返回长度最多为 length－1 字节的字符串。碰到换行符（包括在返回值中）、EOF 或者已经读取了 length－1 字节后停止（看先碰到哪一种情况）。如果没有指定 length，则默认为 1K，或者说 1024 字节。出错时返回 FALSE。

3. int fwrite(resource handle,string string [,int length])

该函数用于写入文件，作用同 int fputs(resource handle,string str,int [length])。

fwrite()把 string 的内容写入文件指针 handle 处。如果指定了 length，当写入了 length 个字节或者写完了 string 以后，写入就会停止。

fwrite()返回写入的字符数，出现错误时则返回 FALSE 。

4. bool fclose(resource handle)

该函数用于关闭一个已打开的文件指针，也就是将 handle 指向的文件关闭。如果成功则返回 TRUE，失败则返回 FALSE。文件指针必须有效，并且是通过 fopen()或 fsockopen()成功打开的。

下面就用以上几个简单的文件操作函数来编写一个简单的文本类型的访客计数器。

程序清单　文档 7-1.php

```
<!--文件 7-1.php：访客计数器-->
<html>
    <head>
        <title>访客计数器</title>
    </head>
    <body>
        <?php
            if(!@$fp=fopen("num.txt","r")){
                //只读方式打开 num.txt 文件
                echo "文件打开失败或找不到 num.txt 文件！<br>";
                echo "num.txt 文件创建成功！<br>";
              }
            @$num=fgets($fp,12);   //读取 11 位数字
            if($num=="")$num=0;
              //如果文件的内容为空，初始化为 0
            $num++;              //浏览次数加一
            @fclose($fp);         //关闭文件
            $fp=fopen("num.txt","w");//只写方式打开 num.txt 文件
            fwrite($fp,$num);    //写入加一后结果
            fclose($fp);          //关闭文件
            echo "您是第".$num."位浏览者!";         //浏览器输出浏览次数
        ?>
    </body>
</html>
```

当上面的程序第一次被运行时，其运行结果如图 7-1（a）所示。



图 7-1(a)　7-1.php 程序的运行结果

当第二次运行时，其运行效果如图 7-1（b）所示。再次运行，则浏览者编号顺次加 1。



图 7-1（b） 7-1.php 程序的运行结果

从这个例子不难发现，制作一个文件类型计数器的基本思路是：

（1）打开一个文件；
（2）读出文件里面的内容（数据）；
（3）数据+1 后再写入该文件；
（4）关闭文件。

由于当文件以可读可写方式打开时，文件的内容同时被清空，所有在做文件计数器时的具体的实际步骤是：

（1）以只读方式打开一个文件；
（2）读出文件里面的内容（数据）；
（3）关闭文件；
（4）再以可读可写方式打开文件；
（5）数据+后再写入该文件；
（6）关闭文件。

另外，PHP 还提供了一些文件操作的函数，现简单列举如表 7-1 所示。

表 7-1 PHP 文件特征函数一览

函数	作用
Basename	返回路径中的文件名部分
Chgrp	改变文件所属的组
chmod	改变文件模式
chown	改变文件的所有者
clearstatcache	清除文件状态缓存
copy	拷贝文件
delete	参见 unlink()或 unset()
dirname	返回路径中的目录部分
disk_free_space	返回目录中的可用空间
disk_total_space	返回一个目录的磁盘总大小

（续表）

函数	作用
diskfreespace	disk_free_space()的别名
fclose	关闭一个已打开的文件指针
feof	测试文件指针是否到了文件结束的位置
fflush	将缓冲内容输出到文件
fgetc	从文件指针中读取字符
fgetcsv	从文件指针中读入一行并解析 CSV 字段
fgets	从文件指针中读取一行
fgetss	从文件指针中读取一行并过滤掉 HTML 标记
file_exists	检查文件或目录是否存在
file_get_contents	将整个文件读入一个字符串
file_put_contents	将一个字符串写入文件
file	把整个文件读入一个数组中
fileatime	取得文件的上次访问时间
filectime	取得文件的 inode 修改时间
filegroup	取得文件的组
fileinode	取得文件的 inode
filemtime	取得文件修改时间
fileowner	取得文件的所有者
fileperms	取得文件的权限
filesize	取得文件大小
filetype	取得文件类型
flock	轻便的咨询文件锁定
fnmatch	用模式匹配文件名
fopen	打开文件或者 URL
fpassthru	输出文件指针处的所有剩余数据
fputcsv	将行格式化为 CSV 并写入文件指针
fputs	fwrite()的别名
fread	读取文件（可安全用于二进制文件）
fscanf	从文件中格式化输入
fseek	在文件指针中定位
fstat	通过已打开的文件指针取得文件信息
ftell	返回文件指针读/写的位置
ftruncate	将文件截断到给定的长度
fwrite	写入文件（可安全用于二进制文件）
glob	寻找与模式匹配的文件路径
is_dir	判断给定文件名是否是一个目录

（续表）

函数	作用
is_executable	判断给定文件名是否可执行
is_file	判断给定文件名是否为一个正常的文件
is_link	判断给定文件名是否为一个符号连接
is_readable	判断给定文件名是否可读
is_uploaded_file	判断文件是否是通过 HTTP POST 上传的
is_writable	判断给定的文件名是否可写
is_writeable	is_writable()的别名
link	建立一个硬连接
linkinfo	获取一个连接的信息
lstat	给出一个文件或符号连接的信息
mkdir	新建目录
move_uploaded_file	将上传的文件移动到新位置
parse_ini_file	解析一个配置文件
pathinfo	返回文件路径的信息
pclose	关闭进程文件指针
popen	打开进程文件指针
readfile	输出一个文件
readlink	返回符号连接指向的目标
realpath	返回规范化的绝对路径名
rename	重命名一个文件或目录
rewind	倒回文件指针的位置
rmdir	删除目录
set_file_buffer	stream_set_write_buffer()的别名
stat	给出文件的信息
symlink	建立符号连接
tempnam	建立一个具有惟一文件名的文件
tmpfile	建立一个临时文件
touch	设定文件的访问和修改时间
umask	改变当前的 umask
unlink	删除文件

详情请查看 PHP 5 的帮助文档。

7.2 目 录 操 作

目录的操作主要是利用相关的目录函数来实现的，下面先来看一下有关的目录函数。

1. string getcwd(void)

取得当前工作目录。

2. bool chdir(string directory)

将当前目录改为 directory。

3. new dir(sting directory)

将输入的目录名转换为一个对象并返回，举例如下：

```
class dir {
dir(string directory)
string path
resource handle
string read(void)
void rewind(void)
void close(void)
}
```

该类含有两个属性和 3 个函数。两个属性为：

- Handle：目录句柄。
- Path：打开目录的路径。

三个函数为：

- read(void)：读取目录。
- rewind(void)：复位目录。
- close(void)：关闭目录。

这三个函数与后面的将要介绍的 readdir()、rewinddir()、closedir()三个函数的作用相同。

4. resource opendir(string path)

打开目录句柄，其中 path 为要打开的目录路径。

5. string readdir(resource dir_handle)

返回目录中下一个文件的文件名。注意，文件名按其在文件系统中的排序返回；dir_handle 为目录句柄的 resource，之前由 opendir()打开；

成功则返回文件名，失败返回 FALSE。

6. void rewinddir(resource dir_handle)

倒回目录句柄，将 dir_handle 指定的目录流重置到目录的开头。其中，dir_handle 为目录句柄的 resource，之前由 opendir()打开。

7. void closedir(resource dir_handle)

关闭目录句柄，关闭由 dir_handle 指定的目录流。其中，dir_handle 为目录句柄的

resource，之前由 opendir()所打开。

8. array scandir(string directory [,int sorting_order])

列出指定路径中的文件和目录，返回一个 array，数组中包含 directory 中的文件和目录。

其中，directory 是要被浏览的目录；sorting_order 是文件的排列顺序，默认的排序顺序是按字母升序排列。如果使用了可选参数 sorting_order（设为 1），则排序顺序是按字母降序排列。

9. bool chroot(string directory)

将当前进程的根目录改变为 directory。

本函数仅在系统支持且运行于 CLI、CGI 或嵌入 SAPI 版本时才能正确工作。此外，本函数还需要 root 权限。

关于目录操作函数的具体应用，下面通过一个简单的例子加以说明。

程序清单　文档 7-2.php

```
<!--文件 7-2.php：目录输出-->
<HTML>
 <HEAD>
     <TITLE>目录输出</TITLE>
 </HEAD>
 <BODY>
     <?php
          $dir=getcwd();     //获取当前路径
          echo getcwd(). "<br>";  //输出当前目录
          $files1=scandir($dir);   //列出指定路径中的文件和目录
           $files2=scandir($dir,1);
           print_r($files1);   //输出指定路径中的文件和目录
           print_r($files2);
           $dir=dir($dir);
           echo gettype($dir)."<br>";
         echo "目录句柄:".$dir->handle."<br>";
         echo "目录路径:".$dir->path."<br>";
         while($entry=$dir->read())
             echo $entry."; <br> ";
             $dir->close();
           if(chdir("c:/windows")){
               echo "当前目录更改成功:c:/windows<br>";
               }else{
                  echo "当前目录更改失败! <br>";
                   }
     ?>
 </BODY>
```

```
</HTML>
```

其运行结果如图 7-2 所示。

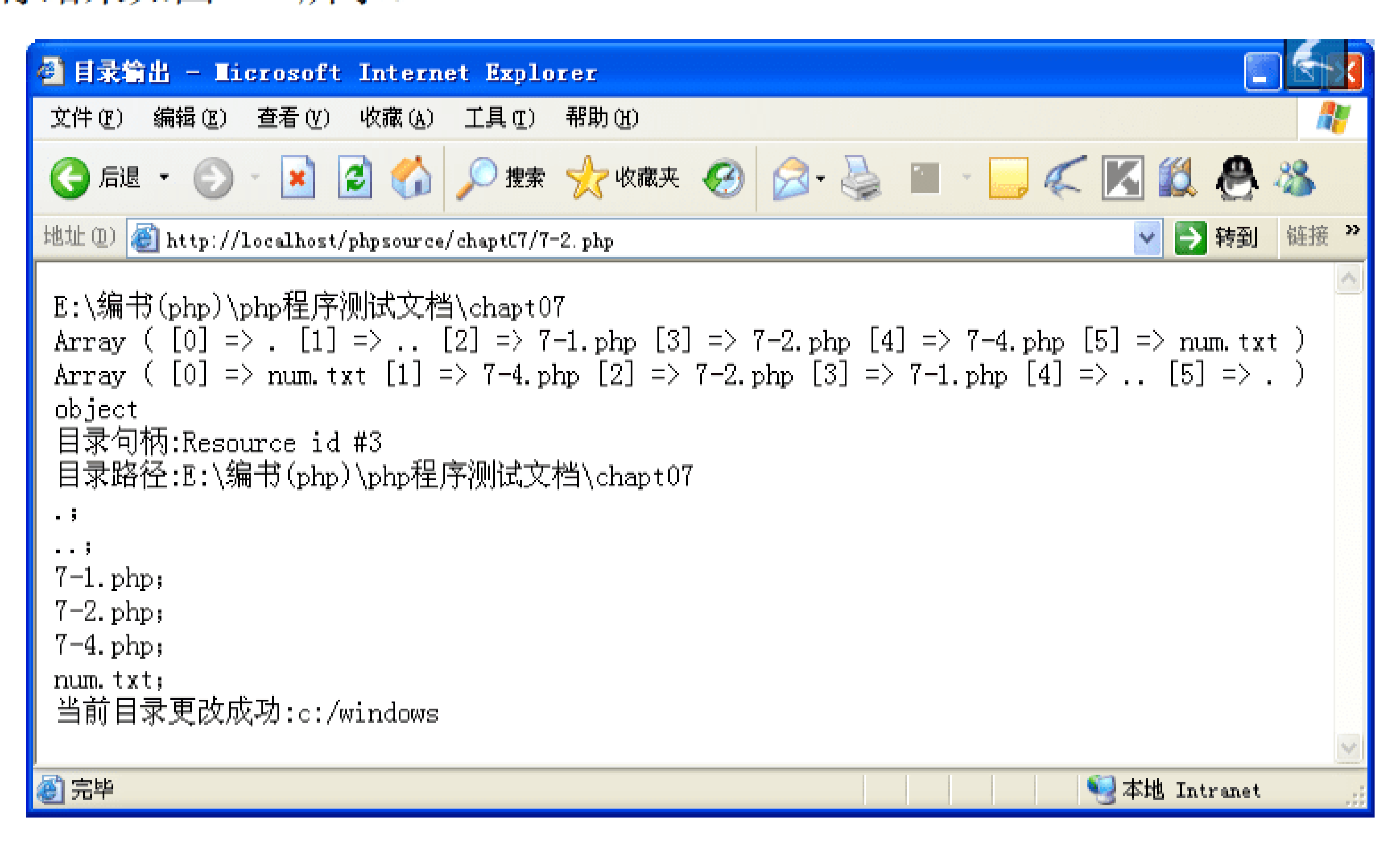


图 7-2　“7-2.php”程序的运行结果

下面再来看一个例子，体会一下目录操作在实际中的具体应用。

程序清单　文档 7-3.php

```
<!--文件 7-3.php：遍历图片显示-->
<HTML>
 <HEAD>
      <TITLE>遍历图片显示</TITLE>
 </HEAD>
 <BODY>
      <?php
           $addr="./image/";
           $dir=dir($addr);
           while($file_name=$dir->read()){
             if($file_name=="." or $file_name==".."){
               }else{
         echo "<img src=".$addr.$file_name." width=40
         height=30>  ";
                }
           }
      ?>
  </BODY>
 </HTML>
```

其运行结果如图 7-3 所示。



图 7-3 “7-3.php”程序的运行结果

提 示

> 通过这个实例，就可以做出企业网站中的宣传图片或商业网站的广告了，只要在更新这些图片的时候，添加删除图片就可以了，这样是不是更简单些呢？

7.3 文件上传的实现

文件上传的功能也是经常使用的，有了文件上传的功能，不仅可以为网站动态添加附件，来完善网页文字编辑比较弱的功能，而且还可以实现网站中相关图片、Flash 动画等的动态更新等等。下面就通过一个实例来了解一下文件上传的基本原理。

程序清单 文档 7-4.php

```
<!--文件 7-4.php：文件上传实例-->
<!--为了能正确运行，请在本文件的同目录下新建一文件夹，名为"upfile"-->
<?php
if($_POST[add]=="上传"){
        //根据现在的时间产生一个随机数
        $rand1=rand(0,9);
        $rand2=rand(0,9);
        $rand3=rand(0,9);
        $filename=date("Ymdhms").$rand1.$rand2.$rand3;
        if(empty($_FILES['file_name']['name'])){
              //$_FILES['file_name']['name']为获取客户端机器文件的原名称
          echo "文件名不能为空";
          exit;
          }
       $oldfilename=$_FILES['file_name']['name'];
       echo "<br>原文件名为：".$oldfilename;
$filetype=substr($oldfilename,strrpos($oldfilename,"."),
  strlen($oldfilename)-strrpos($oldfilename,"."));
       echo "<br>原文件的类型为：".$filetype;
```

```
if(($filetype!='.doc')&&($filetype!='.xls')&&($filetype!='.DOC')&&
($filetype!='.XLS')){
        echo "<script>alert('文件类型或地址错误');</script>";
        echo "<script>location.href='7-5.php';</script>";
        exit;
        }
    echo "<br>上传文件的大小为（字节）：".$_FILES['file_name']['size'];
        //$_FILES['file_name']['size']为获取客户端机器文件的大小，单位为 B
    if($_FILES['file_name']['size']>1000000){
        echo "<script>alert('文件太大，不能上传');</script>";
        echo "<script>location.href='7-5.php';</script>";
        exit;
        }
    echo "<br>文件上传服务器后的临时文件名为：
".$_FILES['file_name']['tmp_name'];
                //取得保存文件的临时文件名（含路径）
    $filename=$filename.$filetype;
    echo "<br>新文件名为：".$filename;
    $savedir="upfile/".$filename;
    if(move_uploaded_file($_FILES['file_name']['tmp_name'],
 $savedir)){
        $file_name=basename($savedir);
    //取得保存文件的文件名（不含路径）
        echo "<br>文件上传成功！保存为：".$savedir;
        }else{
         echo "<script language=javascript>";
         echo "alert('错误，无法将附件写入服务器!\n 本次发布失败！');";
         echo "location.href='7-5.php?';";
         echo "</script>";
         exit;
       }
  }
?>
<html>
<head>
<meta http-equiv="Content-Language" content="zh-cn">
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>==文件上传实例==</title>
<style>
body{font-size:10pt};
td{font-size:10pt};
.style1 {color:#FF0000}
.style2 {
```

```
color:#000000;
font-weight:bold;
}
</style>
</head>
<body>
<div align="center">
</div>
 <form  enctype="multipart/form-data" action="7-5.php" method="post">
 <!--此处一定要有 enctype="multipart/form-data"//-->
 <table width="486" height="103" border="1" align="center"
  cellpadding="0" cells pacing="0" bordercolor="#008080"
  id="AutoNumber1" style="border-collapse:collapse">
  <tr bgcolor="#CCCCCC">
  <td height="30" colspan="2" align="right"><div align=
  "center" class="style2">文件上传实例</div>     </td>
 </tr>
 <tr>
  <td width="103" height="45" align="right"><div align="center">
   <span class="style1">*</span>文件上传地址：</div></td>
  <td width="377" height="45"><input type="file" name="file_name">
   （大小（2M 为宜）</td>
 </tr>
 <tr>
  <td height="33" align="right" colspan="2">
  <p align="center"><input type="submit" value="上传" name="add">
        <input type="reset" value="重置"
   name="B2"></td>
 </tr>
</table>
</form>
</body>
</html>
```

该程序的运行效果如图 7-4（a）、图 7-4（b）所示。

通过上面的例子，不难发现，文件上传的基本原理是：

（1）从“客户机文件”到“服务器临时文件夹”；

（2）从“服务器临时文件夹”到“服务器上传文件夹”。

期间，从客户机到服务器的上传过程中要通过一系列的验证，比如文件的类型、大小等是否符合要求，从服务器临时文件夹到服务器上传文件夹的复制转移过程中要给文件重新命名等。

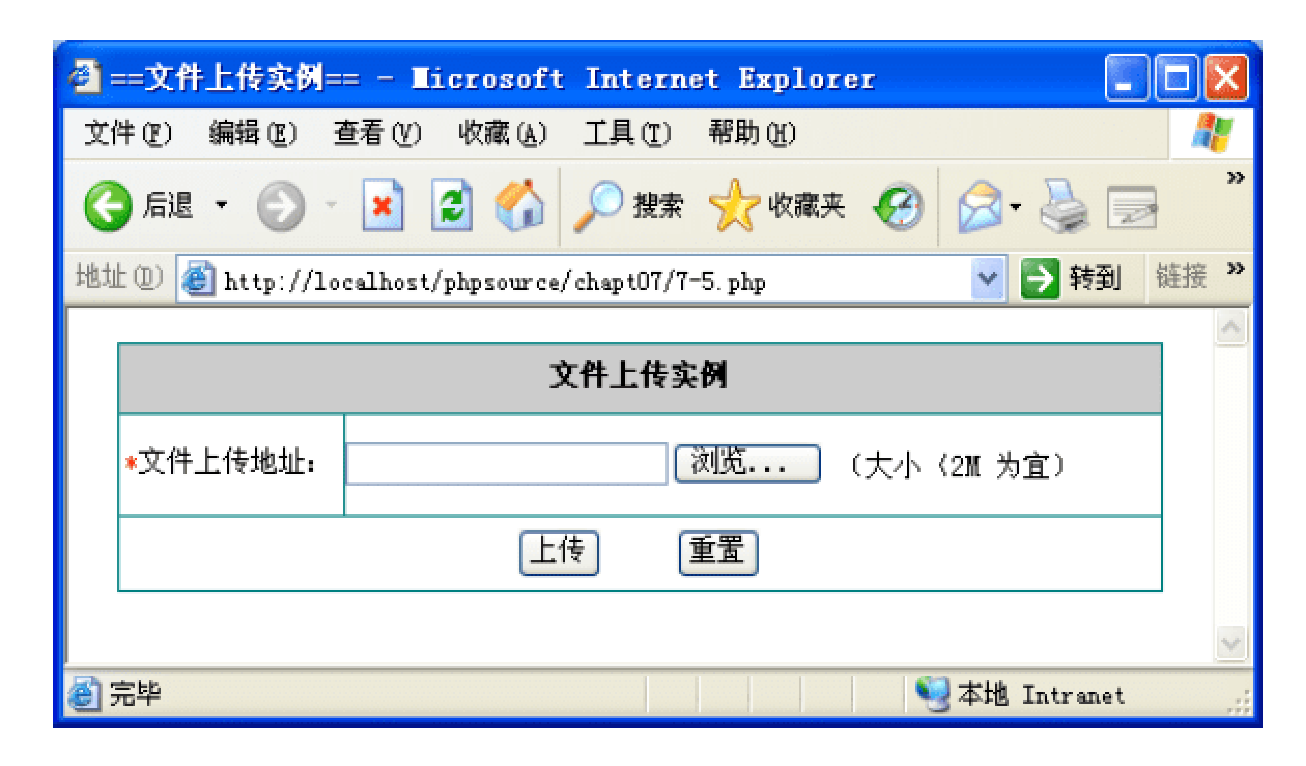


图 7-4（a）　7-4.php 程序的运行结果（未点“上传”按钮前）

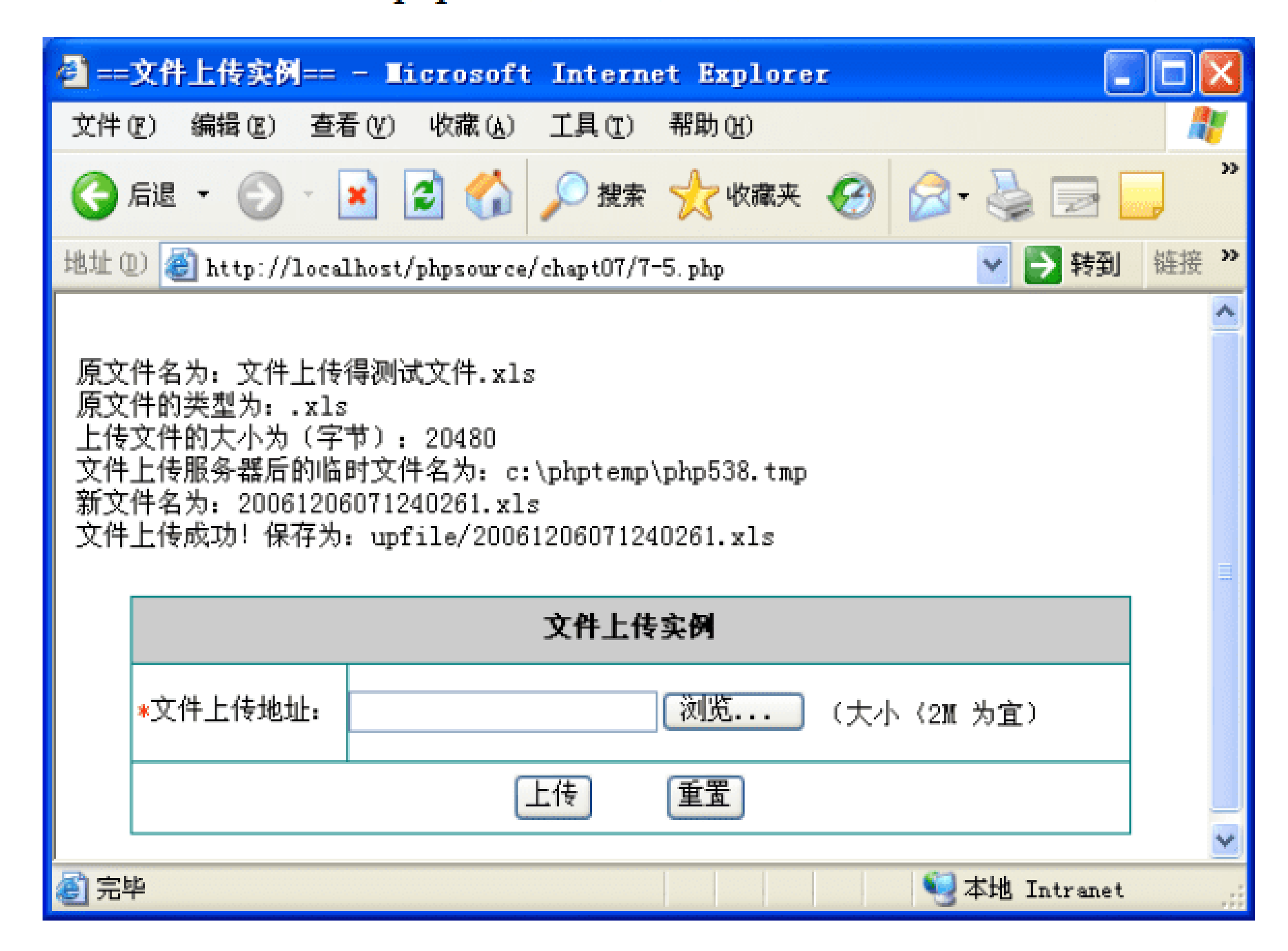


图 7-4（b）　7-4.php 程序的运行结果（点“上传”按钮后）

注 意

“〈form enctype="multipart/form-data"......〉”是一个标记，要实现文件的上传，必须指定为 multipart/form-data，否则服务器将无法判断程序是如何运行的。

上面测试了上传一个文件的例子，那么要上传多个文件应当如何处理呢？PHP 支持同时上传多个文件，并能将它们的信息自动以数组的形式组织起来。要完成这项功能，需要在 HTML 表单中对文件上传使用和多选框与复选框相同的数组式提交语法，可以用下面的方法来处理。

提交的表单可以写成：

```
<form action="file-upload.php" method="post" enctype="multipart/form-data">
  Send these files:<br>
  <input name="userfile[]" type="file"><br>
  <input name="userfile[]" type="file"><br >
  <input type="submit" value="Send files">
</form>
```

当以上表单被提交后，数组$_FILES['userfile']，$_FILES['userfile']['name'] 和 $_FILES['userfile']['size'] 将被初始化。如果 register_globals 的设置为 on，则和文件上传相关的全局变量也将被初始化。所有这些提交的信息都将被储存到以数字为下标的数组中，接受使用即可。

7.4 案例 7 利用文件操作实现支持头像上传的用户注册程序

设计一简单的用户注册和登录界面，将用户注册信息记入文本文件，并实现用户登录与上传个人照片的功能。

7.4.1 案例分析

仔细分析本案例，实际上就是文件操作和文件上传的综合应用，具体制作步骤如下：

（1）设计一简单的用户注册界面，含有上传头像的表单项；

（2）新建一接收用户注册信息的文件，本文件所要完成的主要功能为有：接收注册的用户名、登录口令和头像文件，实现头像（图片）的上传，把用户名、登录口令和上传后的头像文件信息记录到一文本文件中；

（3）设计一简单的用户登录界面（将用户注册的界面稍加修改即可）；

（4）新建一接收用户登录信息的文件，本文件所要完成的主要功能有：接收登录用户的用户名和登录口令，从记录注册信息的文本文件中读出注册用户信息，检查接收的登录用户名是否存在以及登录口令是否正确，若正确则输出登录成功信息和头像图片，否则，输出错误信息。

7.4.2 实施步骤

通过对本案例的具体分析，可以依次编写下面的文件和代码。

1. 制作注册页面

制作一用户注册界面“login.html”，具体代码如下：

程序清单 文档 login.html

```
<!--文件 login.html：用户注册-->
<html>
<head>
<title>用户注册</title>
</head>
<body>
<form enctype="multipart/form-data" name="form1" method="post"
action="login_action.php">
<table width="329" height="150" border="0" align="center"
   cellpadding="0" cellspacing="1" bgcolor="#0000FF">
  <tr>
```

```
   <td colspan="2" align="center" bgcolor="#FFFFFF">用 户 注 册</td>
  </tr>
  <tr>
    <td width="82" align="right" bgcolor="#FFFFFF">用户名:</td>
    <td width="244" align="left" bgcolor="#FFFFFF">
      <input type="text" name="user_name" size="16">    </td>
  </tr>
  <tr>
    <td align="right" bgcolor="#FFFFFF">登录口令:</td>
    <td align="left" bgcolor="#FFFFFF"><input type="password"
name="user_pw1" size="16"></td>
 </tr>
  <tr>
    <td align="right" bgcolor="#FFFFFF">登录口令:</td>
    <td align="left" bgcolor="#FFFFFF"><input type="password"
      name="user_pw2" size="16"></td>
  </tr>
  <tr>
    <td align="right" bgcolor="#FFFFFF">头像:</td>
    <td align="left" bgcolor="#FFFFFF"><input type="file"
name="pic_name"></td>
  </tr>
  <tr>
   <td colspan="2" align="center" bgcolor="#FFFFFF">
<input type="submit" name="Submit" value="注册">  
       <input type="reset" name="Submit2" value="重置"></td>
  </tr>
</table>
</form>
</body>
</html>
```

这是一个静态的 HTML 页面，其运行效果也非常简单，如图 7-5 所示。

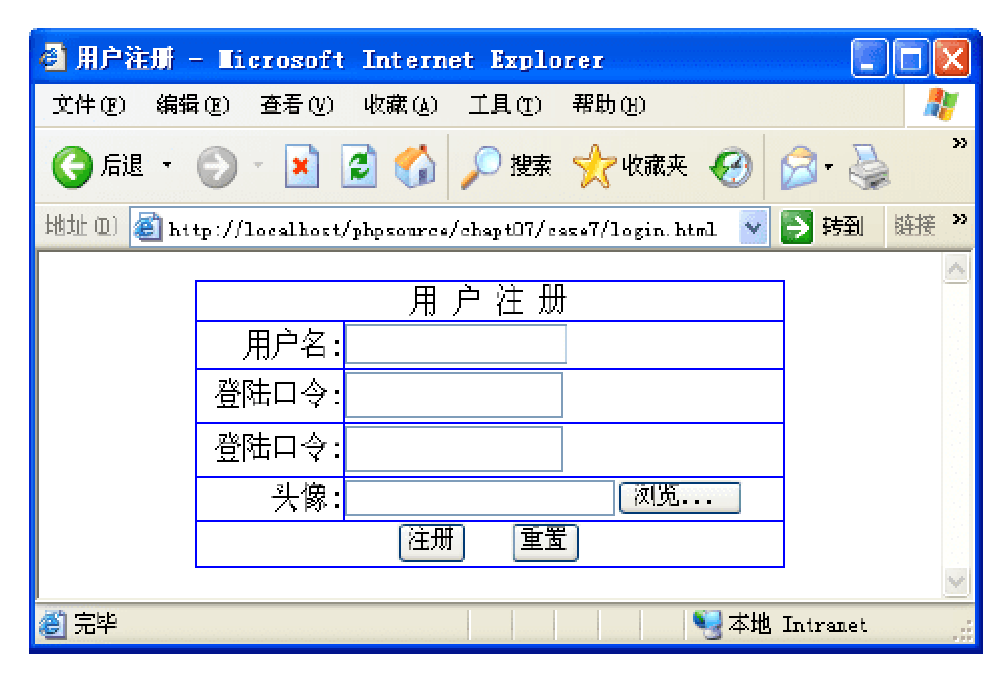


图 7-5　login.html 程序的运行效果

2. 接收注册信息

编写一接收用户注册信息，并使用文本写入的方法记录注册信息的文件“login_action.php”，其具体代码如下：

程序清单 文档 login_action.php

```
<!--文件 login_action.php：用户注册-->
<html>
<head>
<title>用户注册</title>
</head>
<body>
 <?php
    //接收标单数据
 $user_name=$_POST["user_name"];
 $user_pw1=$_POST["user_pw1"];
 $user_pw2=$_POST["user_pw2"];
     //标单数据基本性验证，省略用户名重名验证
 if($user_name=="" or $user_pw1=="" or $user_pw2=="" or $_FILES['pic_name']['name']==""){
 echo "用户名、两次输入的登录口令和头像全不能为空，请<a href=login.html>返回</a>重新填写";
 exit;
 }
      if($user_pw1!=$user_pw2){
 echo "您两次输入的登录口令不匹配，请<a href=login.html>返回</a>重新填写";
      }
        //生成自动文件名
        $rand1=rand(0,9);
        $rand2=rand(0,9);
        $rand3=rand(0,9);
        $filename=date("Ymdhms").$rand1.$rand2.$rand3;
        //头像上传
        $oldfilename=$_FILES['pic_name']['name'];
        $filetype=substr($oldfilename,strrpos($oldfilename,"."),strlen($oldfilename)-strrpos($oldfilename,"."));
        if(($filetype!='.gif')&&($filetype!='.GIF')&&($filetype!='.jpg')&&($filetype!='.JPG')){
           echo "<script>alert('文件类型或地址错误');</script>";
          echo "<script>location.href='login.html';</script>";
           exit;
           }
```

```
33:       if($_FILES['pic_name']['size']>1000000){
34:           echo "<script>alert('文件太大，不能上传');</script>";
35:          echo "<script>location.href='login.html';</script>";
36:          exit;
37:           }
38:                   //取得保存文件的临时文件名（含路径）
39:       $filename=$filename.$filetype;
40:      $savedir=$filename;
41:       if(move_uploaded_file($_FILES['pic_name']['tmp_name'],
    $savedir)){
42:        $file_name=basename($savedir);    //取得保存文件的文件名（不含路径）
43:           }else{
44:             echo "<script language=javascript>";
45:             echo "alert('错误，无法将附件写入服务器!\n 本次发布失败！');";
46:             echo "location.href='login.html';";
47:             echo "</script>";
48:             exit;
49:         }
50: //记录注册信息到 user_info.txt 文件中
51: $file=fopen("user_info.txt","a");
52: flock($file,LOCK_EX);
53:     $string=$user_name."\n";
54: fputs($file,$string);
55: $string=$user_pw1."\n";
56: fputs($file,$string);
57: $string=$filename."\n";
58: fputs($file,$string);
59:     fputs($file,"\n");
60:     echo "恭喜您，注册成功！<br>";
61: echo "用户名：".$user_name."<br>";
62: echo "登录口令：".$user_pw1."<br>";
63: echo "头像：<img src=".$filename." align=middle><br>";
64: echo "请<a href=land.html>登录</a>";
65: ?>
66: </body>
67: </html>
```

分析上面的代码，不难发现，本代码有两个难点，其一就是文件的上传（20 行~49 行），其二就是向一文本文件中追加记录用户注册信息，并在记录每条信息后换行（51 行~59 行）。当全部且正确填写注册信息后，其运行效果图如图 7-6 所示。

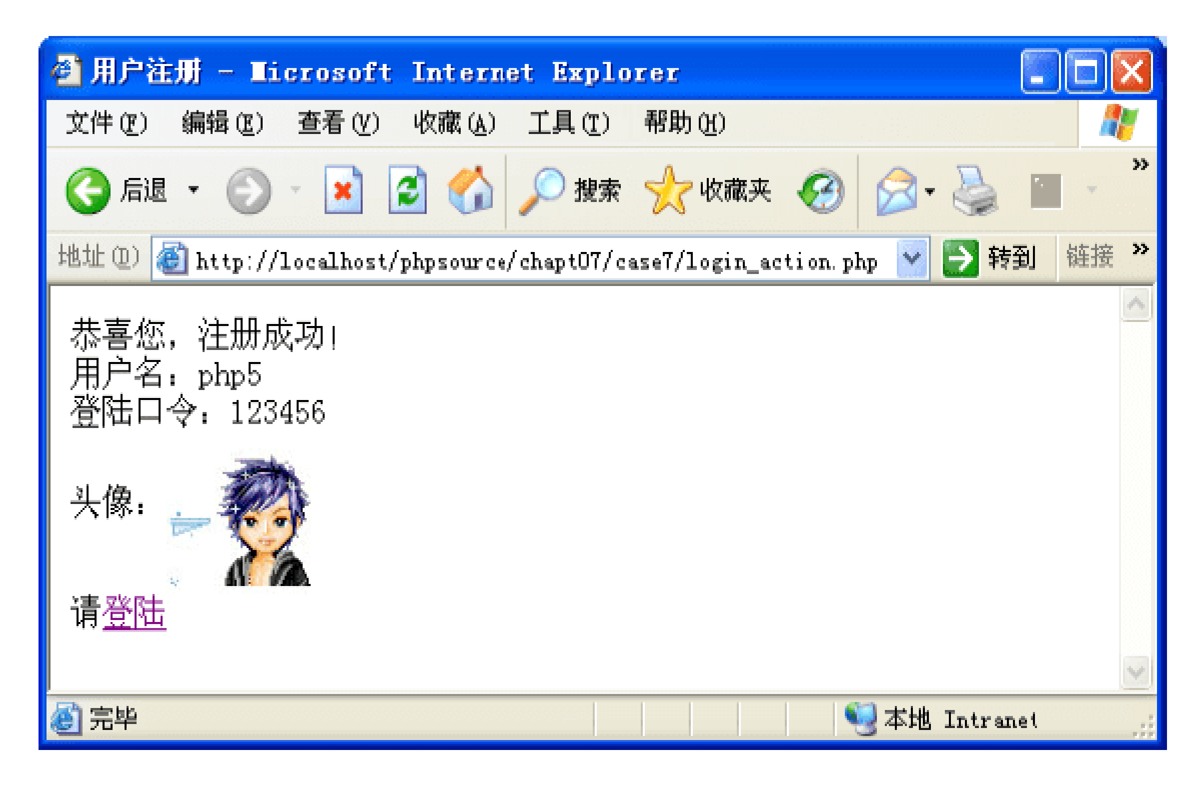


图 7-6 login_action.php 程序的运行效果

3. 设计用户登录界面

建立一个用户登录表单“land.html”，也就是单击图 7-6 中的“登录”链接所调转到的页面。其具体代码如下：

程序清单 文档 land.html

```
<!--文件 land.html：用户登录-->
<html>
<head>
<title>用户登录</title>
</head>

<body>
<form name="form1" method="post" action="land_action.php">
<table width="297" height="105" border="0" align="center"
cellpadding="0" cellspacing="1" bgcolor="#0000FF">
<tr>
    <td colspan="2" align="center" bgcolor="#FFFFFF">用 户 登 陆 </td>
  </tr>
  <tr>
    <td width="85" align="right" bgcolor="#FFFFFF">用户名:</td>
    <td width="209" align="left" bgcolor="#FFFFFF">
      <input type="text" name="user_name" size="16">    </td>
  </tr>
  <tr>
    <td align="right" bgcolor="#FFFFFF">登录口令:</td>
  <td align="left" bgcolor="#FFFFFF"><input type="password"
  name="user_pw" size="16"></td>
  </tr>
  <tr>
    <td colspan="2" align="center" bgcolor="#FFFFFF">
```

```
<input type="submit" name="Submit" value="登录">  
       <input type="reset" name="Submit2" value="重置"></td>
  </tr>
</table>
</form>
</body>
</html>
```

这也是一个静态的 HTML 页面，其运行的效果图如图 7-7 所示。

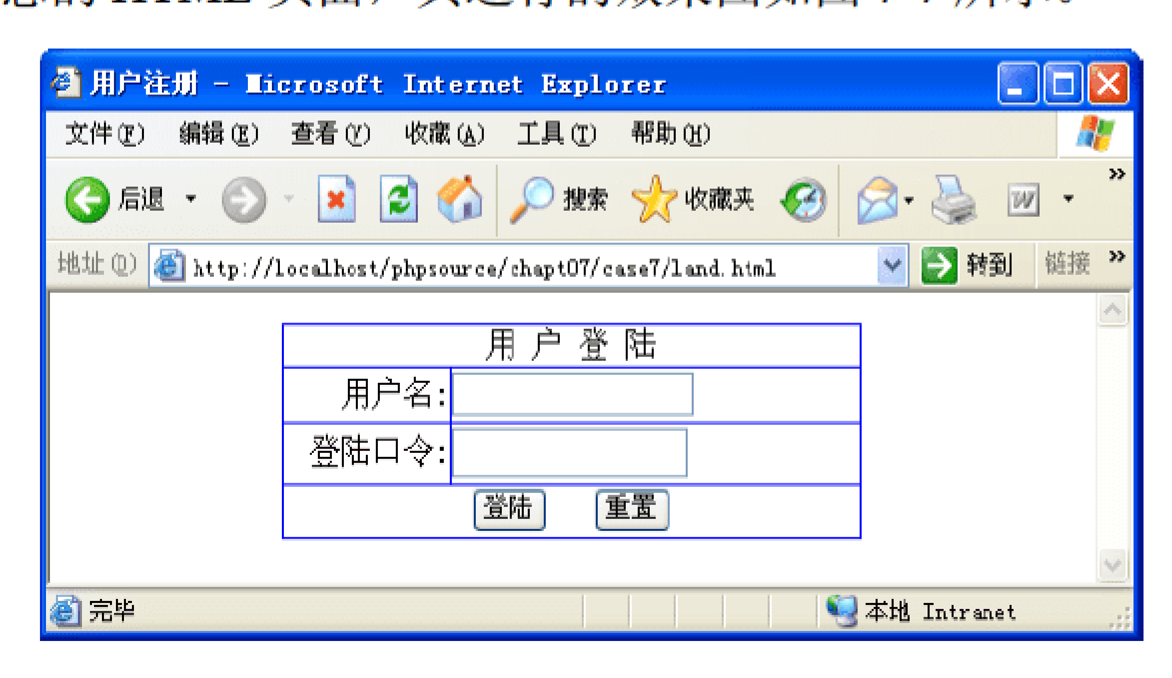


图 7-7　land.html 程序的运行效果

4. 接收用户登录信息

建立接收用户登录信息的页面，并进行判断用户名和密码是否正确，输出相应信息或头像的页面。其基本思路和具体的实施步骤应该是：

① 接受登录用户的用户名和登录口令，并做相应的处理（连接上换行符）；
② 打开文本文件，并锁定，指针定位于第一行；
③ 读出指针所指行的数据，判断是否为登录的用户名；
④ 若③正确，指针下移；
⑤ 读出指针所指行的数据，判断是否为登录口令；
⑥ 若⑤正确，输出登录正确信息和用户头像；
⑦ 若⑤不正确，输出登录错误信息，指针后移两行；
⑧ 若③不正确，指针下移三行；
⑨ 若指针不为空，执行③；
⑩ 如指针为空，输出登录错误信息。

根据具体的实施步骤，可以编写出如下代码：

程序清单　文档 login_action.php

```
<!--文件 login_action.php：用户登录-->
<html>
<head>
<title>用户登录</title>
</head>
```

```
<body>
 <?php
 $user_name=$_POST["user_name"];
 $user_pw=$_POST["user_pw"];
 $user_name2=$user_name."\n";
 $user_pw2=$user_pw."\n";
 $tag=0;    //登录是否成功标记位 0-部成功；1-成功
     $file=@fopen("user_info.txt","r");
 if($file){
     while(!feof($file)){      //循环，文件指针下移
         $buffer=fgets($file,4096); //取一行文本
         if($buffer==$user_name2){
             @next($file);
             $buffer=fgets($file,4096);
             if($buffer==$user_pw2){
                 echo "您输入的用户名密码正确,登录成功！<br>";
                 @next($file);
                 $image=fgets($file,4096);
                 echo "您的头像：<img src=".$image." align=middle><br>";
                 $tag=1;
                 break;
             }else{
                 @next($file);
                 @next($file);
             }
         }else{
             @next($file);
             @next($file);
             @next($file);
         }
     }
     if($tag==0){
         echo "您输入的用户名和密码不正确，请<a href=land.html>返回</a>重新输入";
     }
     fclose($file);
 }
 ?>
 </body>
 </html>
```

不难发现，本程序得以实现的难点是对文本文件的逐行读取。当正确填写用户登录信息后，其运行效果如图 7-8 所示。



图 7-8　land_action.php 程序的运行效果

7.5　本 章 小 结

本章中着重讲解了 PHP 5 对文件和目录的操作以及文件上传的实现。通过学习，我们应该能利用 PHP 5 中对文件和目录的相关操作，实现诸如文本计数器、目录和文件的遍历、不同类型文件的上传等，进而可以做出利用文本操作实现用户注册信息的记录和登录信息的校验等。

7.6　练 习 题

1. PHP 5 中文件打开的方式有几种？有什么样的文件操作权限？
2. PHP 5 文本文件操作可否实现计数器的功能？若能，如何实现？
3. PHP 5 中要实现文件的上传需要哪几步？具体代码是什么？
4. PHP 5 中用哪些函数可以实现从文本文件中逐行读出数据？

CHAPTER 8

MySQL 数据库安装与使用

本章重点

- MySQL 数据库的初始安全设置
- 结构化查询语言 SQL
- phpMyadmin 的使用

PHP + MySQL

8.1　MySQL 数据库系统简介

8.1.1　Web 开发与数据库

动态网站开发离不开数据存储，数据存储则离不开数据库。在前面的章节中，曾做过一个例子，将注册用户的信息存储在一个文本文件中，可以在以后取用，这使得网站可以增加很多交互性因素。但是文本文件并不是存储数据的最理想方法。数据库技术的引入给网站开发带来巨大的飞跃。

数据库技术是计算机技术中的重要部分，在软件开发等领域起着至关重要的作用。这里用比较通俗的语言描述一下什么是数据库。

所谓数据库，可以理解为用来存储信息的“仓库”。而“信息”就是要存储下来的一些数据，比如用户的姓名、年龄，产品的价格、简介，某一个日期时间甚至图像等等。总之一切可以在计算机中存储下来的数据都可以通过各种方法存储到数据库中。

信息并不是杂乱无章的直接放入数据库，而是以二维表的形式组织起来，一条一条存储于表中，这和日常生活中经常用到的各种表格形式上是一致的。表中的每一条信息称为一条“记录”。一个数据库中可以有若干张表，每张表中又可以存放若干条记录。比如前面讲到的用户注册程序，每一个用户的信息，比如用户名、密码、头像等，就可以作为一条记录，存储在一张表中。

每张表都有自己的“表头”。比如设计一张表格，用来统计学生信息，那么把要收集的学生的信息分成几个栏目，这些栏目就是“表头”，在数据库技术中，称为“字段”。表 8-1 是一张学生基本信息表，这里面“学号”、“姓名”、“性别”、“年龄”就是字段，“张三”就是第一条记录的“姓名”字段值，“20”就是第三条记录的“年龄”字段值。表中横向的多个字段值组成了一条记录，多条记录构成了一张数据表。

表 8-1　二维表示意图

学号	姓名	性别	年龄
001	张三	男	18
002	李四	女	19
003	王五	男	20
…	…	…	…

这样就简要说明了数据库、表、字段和字段值这几个概念。实际上这些概念远比这里介绍的复杂得多。可以参考数据库技术的相关书籍来进一步理解。

把数据以这种形式存放在数据库中能带来什么好处呢？采取数据库技术可以给数据的存储和检索带来巨大好处，主要可以归纳为以下 4 点：

（1）数据存储集约化，最大限度节省存储空间。

（2）数据库专门的检索引擎能够极大提高数据检索速度。

（3）数据库结构化查询语言（SQL）给数据管理带来了极大便利。

（4）可以方便地对数据进行查询、增加、删除、修改。

数据库系统就其根本就是一个软件系统。通过这个软件系统可以对大量数据进行存储和管理。当前市场上的数据库有几十种，其中有如 Oracle、SQL Server 等大型网络数据库，也有如 Access、VFP 等小型桌面数据库。对于网站开发而言，一般来说中小型数据库系统就能满足要求。MySQL 就是当前 Web 开发中尤其是 PHP 开发中使用最为广泛的数据库。

8.1.2 MySQL 数据库简介

MySQL 是 MySQL AB 公司开发的一种开放源代码的关系型数据库管理系统(RDBMS)，它使用最常用的数据库管理语言——结构化查询语言（SQL）进行数据库管理。由于 MySQL 是开放源代码的，因此任何人都可以在 General Public License 的许可下下载并根据个性化的需要对其进行修改。MySQL 因其速度、可靠性和适应性而备受关注。大多数人都认为在不需要事务化处理的情况下，MySQL 是管理内容最好的选择。

MySQL 关系型数据库于 1998 年 1 月发行第一个版本。它使用系统核心提供的多线程机制提供完全的多线程运行模式，提供了面向 C、C++、Eiffel、Java、Perl、PHP、Python 等编程语言的编程接口，支持多种字段类型并且提供了完整的操作符。

2001 年 MySQL 4.0 版本发布。在这个版本中提供了新的特性：新的表定义文件格式、高性能的数据复制功能、更加强大的全文搜索功能等。目前，MySQL 已经发展到 MySQL 5.1，功能和效率方面都得到了更大的提升。

大概是由于 PHP 开发者特别衷情于 MySQL，因此才在 PHP 中建立了完美的 MySQL 支持。在 PHP 中，用来操作 MySQL 的函数一直是 PHP 的标准内置函数。开发者只需要用 PHP 写下短短几行代码，就可以轻松连接到 MySQL 数据库。PHP 还提供了大量的函数来对 MySQL 数据库进行操作。用 PHP 操作 MySQL 数据库极为简单和高效，这也使得 PHP+MySQL 成为当今最为流行的 Web 开发语言与数据库搭配之一。

当然，PHP 支持的数据库远远不止 MySQL 一种。根据 PHP 官方提供的资料，PHP 支持几乎全部当前主流的数据库。但是 PHP 和 MySQL 的搭配无论从性能上还是到易用性上都毫无疑问的成为了开发者的首选。此外，还有一个重要原因就是 PHP 和 MySQL 都是免费和开放源代码的，并且都有良好的跨平台特性，这使得搭建 Web 服务器的成本几乎为零，而且开发出来的程序具有可移植性。这些都是吸引开发者的重要原因。

本章中就先介绍 MySQL 数据库的一些基础知识，在第 9 章中将介绍 PHP 连接到 MySQL 数据库进行数据操作的方法。

8.2 MySQL 的安装与初始化设置

虽然 MySQL 5.1 已经发布，但本书中以 MySQL 5.0.18 这个目前相对稳定的版本来介绍。其他版本的 MySQL 安装方法也大致相同。

8.2.1　下载 MySQL 安装

首先要下载 MySQL 的安装包。可以直接从 MySQL 的官方网站来下载，网址 http://www.mysql.com。也可以通过国内站点来下载，如 http://www.mysql.cn。由于本书中的例子都是在 Windows 平台下进行开发和调试，因此要下载 Windows 平台下的 MySQL 安装包。如一个名为“mysql-5.0.18-win32.zip”的压缩包，通过压缩包的名字就可以看出，这个 MySQL 版本为 5.0.18，并且适用于 Windows 32 位平台。

8.2.2　安装 MySQL

安装步骤如下：

（1）首先解压缩，得到一个“setup.exe”的安装文件，双击此文件启动安装程序。稍候之后会出现软件安装欢迎界面，如图 8-1 所示。

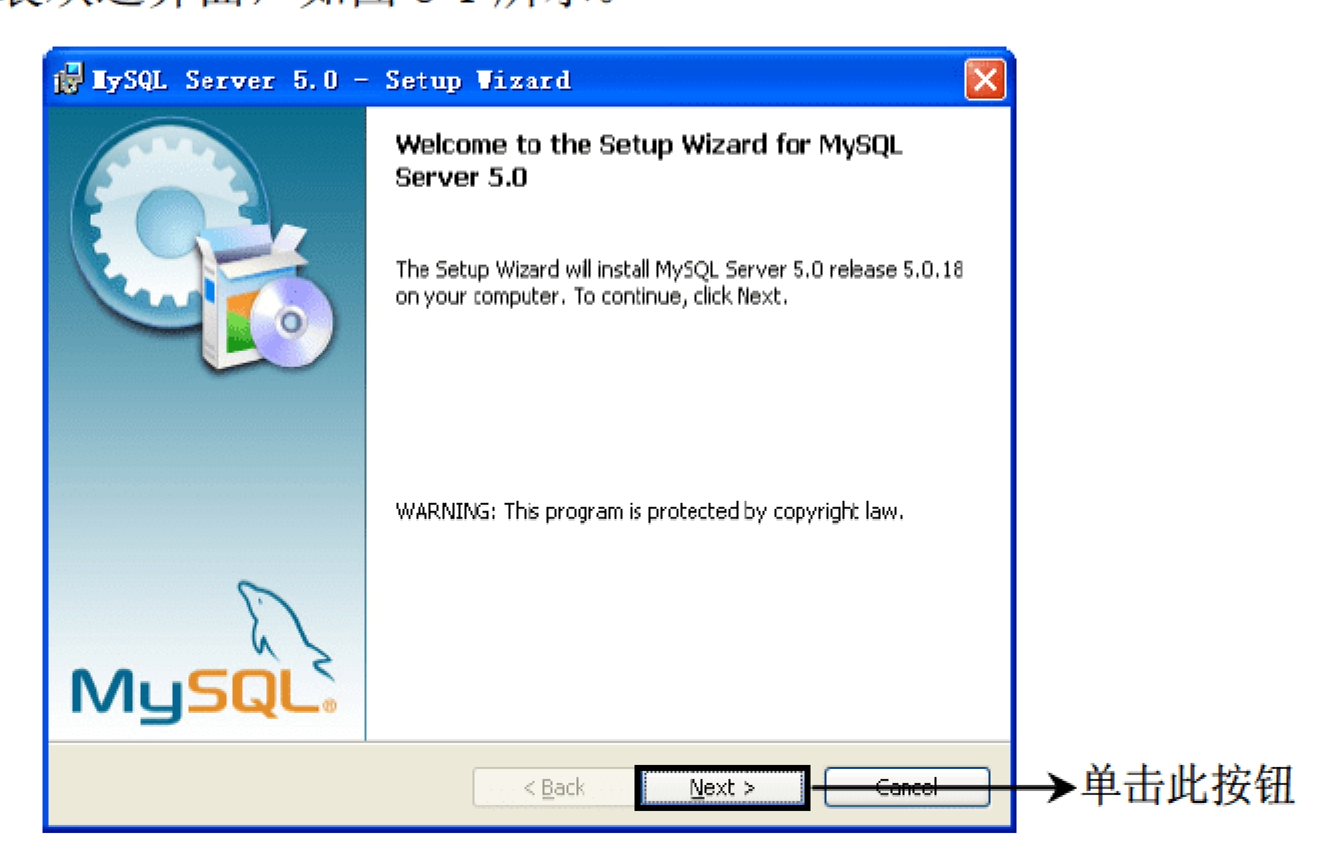


图 8-1　MySQL 数据库安装

（2）单击 Next 按钮，出现安装类型选择窗口，如图 8-2 所示。默认选项为 Typical 即“典型”。这里建议选择第三种 Custom 即“自定义”。因为选择此项才能手工指定安装目录，否则将会安装到默认目录。

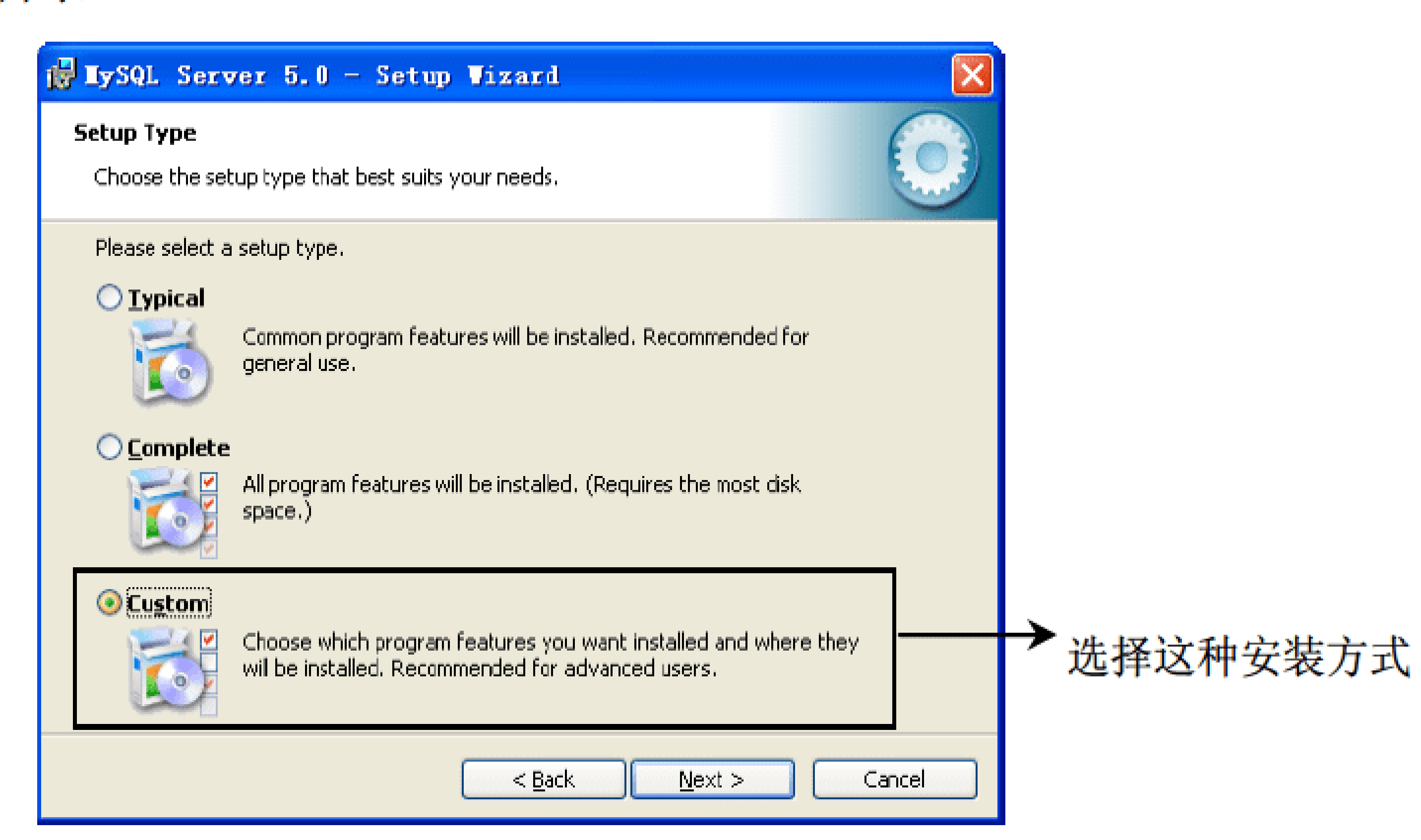


图 8-2　安装类型选择

（3）单击 Next 按钮，出现“自定义安装”界面。在这里可以选择要安装的组件。因为安装全部组件也仅需要 23MB 的空间，因此建议不作改动，保持默认。窗口下方为安装位置，默认为“C:\Program Files\MySQL\MySQL Server5.0\”，可以看出这个目录较为复杂，不便于将来使用，可以单击 Change 按钮，另外选择一个目录。例如选择“C:\mysql5\”作为安装目录。如图 8-3 所示。

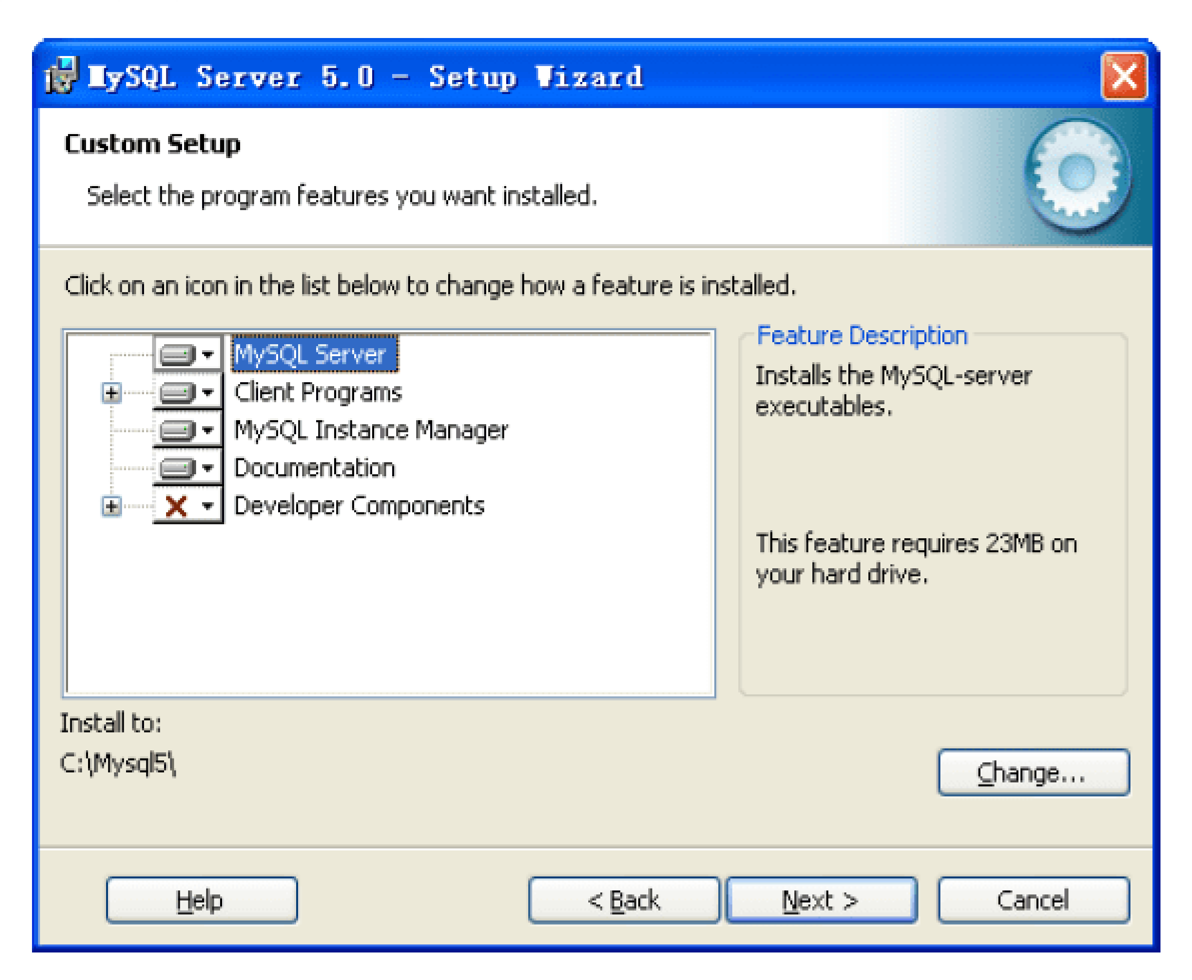


图 8-3　安装组件与目录选择

（4）继续单击 Next，将出现“准备就绪，确认安装”的窗口，确认无误后单击 Install 按钮，此时出现安装进度条，如图 8-4 所示。安装进度完成后，会出现 MySQL 注册窗口，提示创建一个 MySQL 通行证或者登录 MySQL 通行证。可以选择 Skip Sign-up 直接跳过这一环节，如图 8-5 所示。

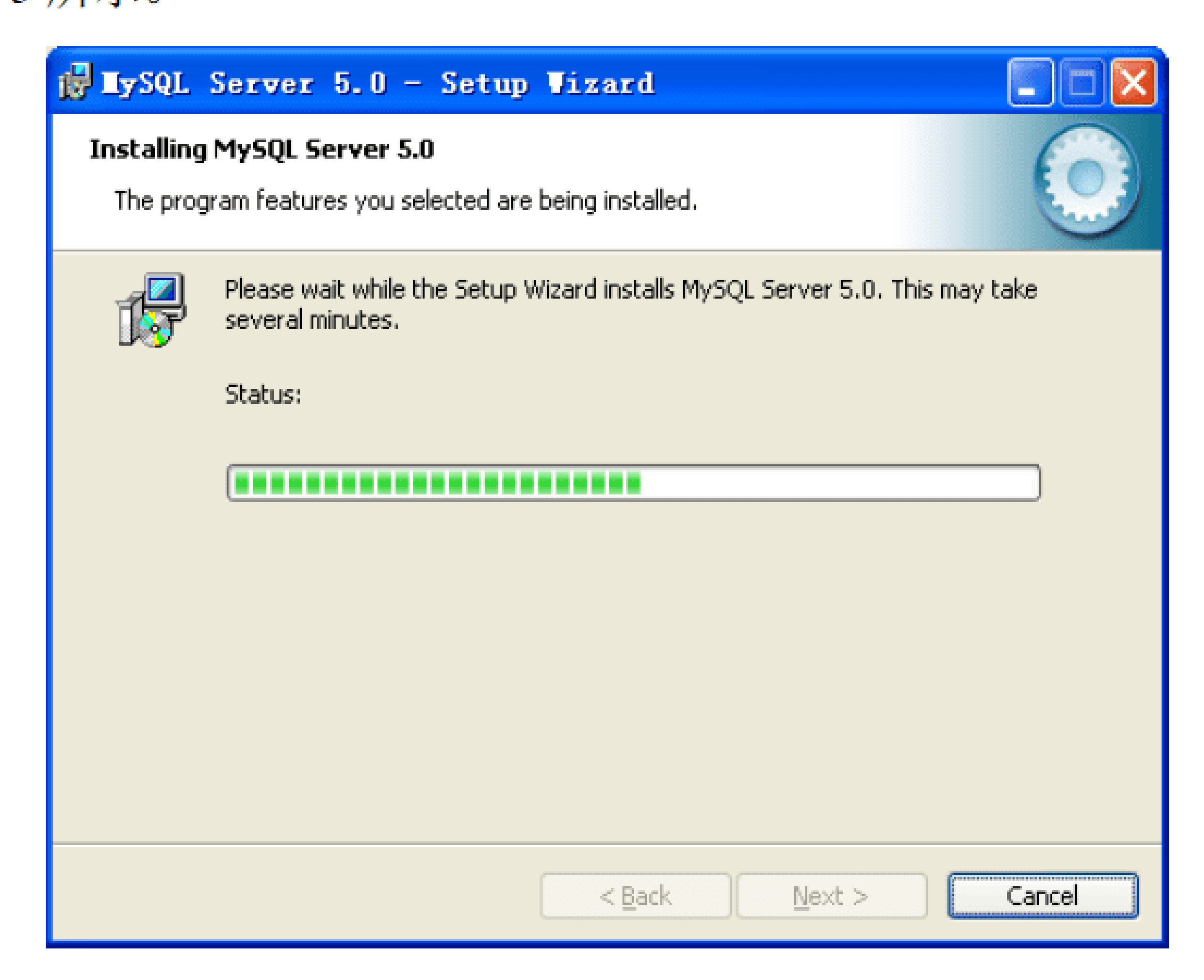


图 8-4　MySQL 安装进度

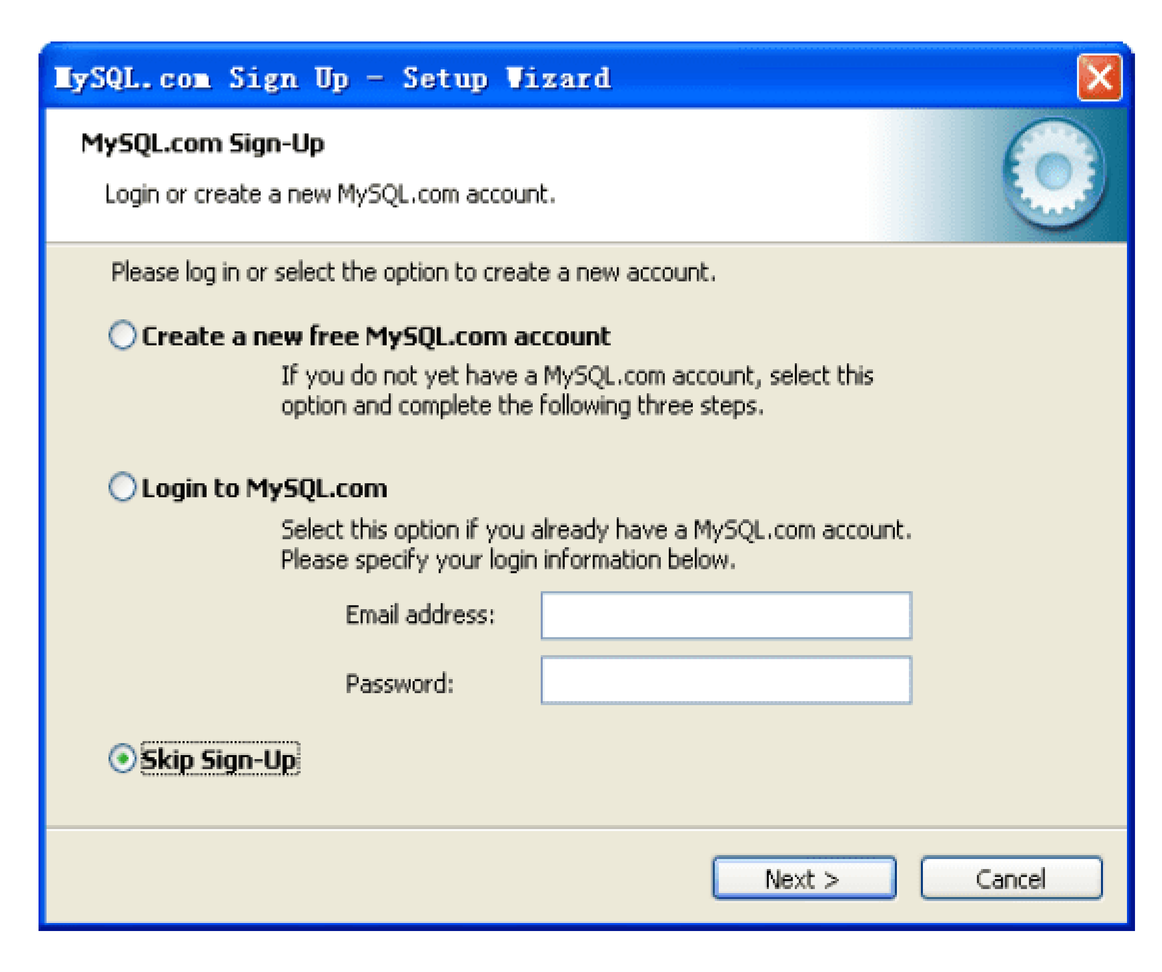


图 8-5　MySQL 联机注册

（5）单击 Next 按钮，出现安装完毕窗口。这时有一个复选框，询问是否进行 MySQL 服务器配置。建议保持选中，并单击 Finish 按钮，这时可以启动 MySQL 服务器配置向导。

（6）单击 Next 按钮，出现服务器配置类型窗口，保持默认的 Detailed Configuration，单击 Next 按钮，这时出现三个选项，可以选择 Developer Machine 或 Server Machine。这里选择 Developer Machine。单击 Next 按钮，出现选择数据库用途界面，可以选择 Multifunctional Database 即“多功能数据库”。单击 Next 按钮，出现 InnoDB 存放位置窗口，保持默认直接点 Next 按钮，出现并发连接数选择窗口，如图 8-6 所示。这里第一个选项表示最大连接数为 20，第二个选项表示最大连接数为 500，第三个为自定义连接数。在学习阶段，选择第一个，20 个连接数足够使用。如果是真正配置 Web 服务器，可以根据需要选择更大的连接数。

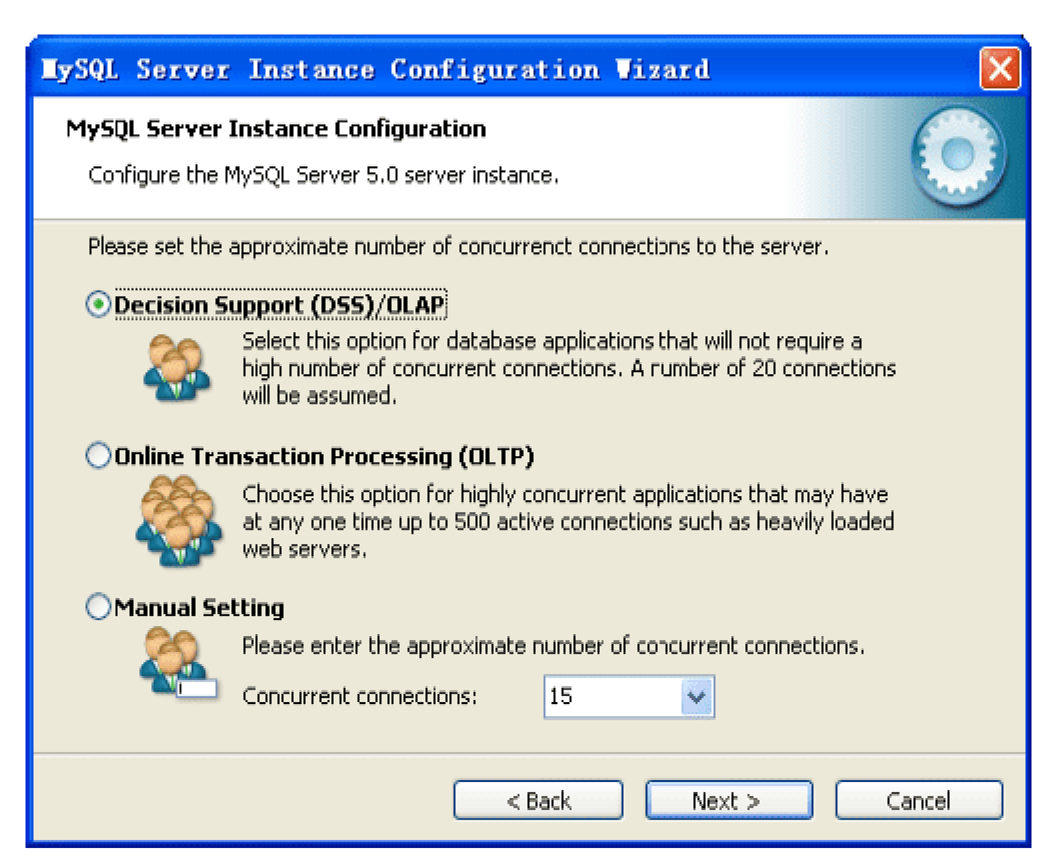


图 8-6　最大连接数选择

（7）单击 Next，出现端口选择窗口。这里就是 MySQL 数据库的服务端口，如果没有特殊需要，直接使用默认的 3306 端口即可。单击 Next，出现字符集设置窗口，如图 8-7 所示。第一个选项为使用默认字符集，也就是把“Latin1”作为默认字符集，第二个选项为把 UFT-8 设置为默认字符集，第三个选项为自定义字符集。

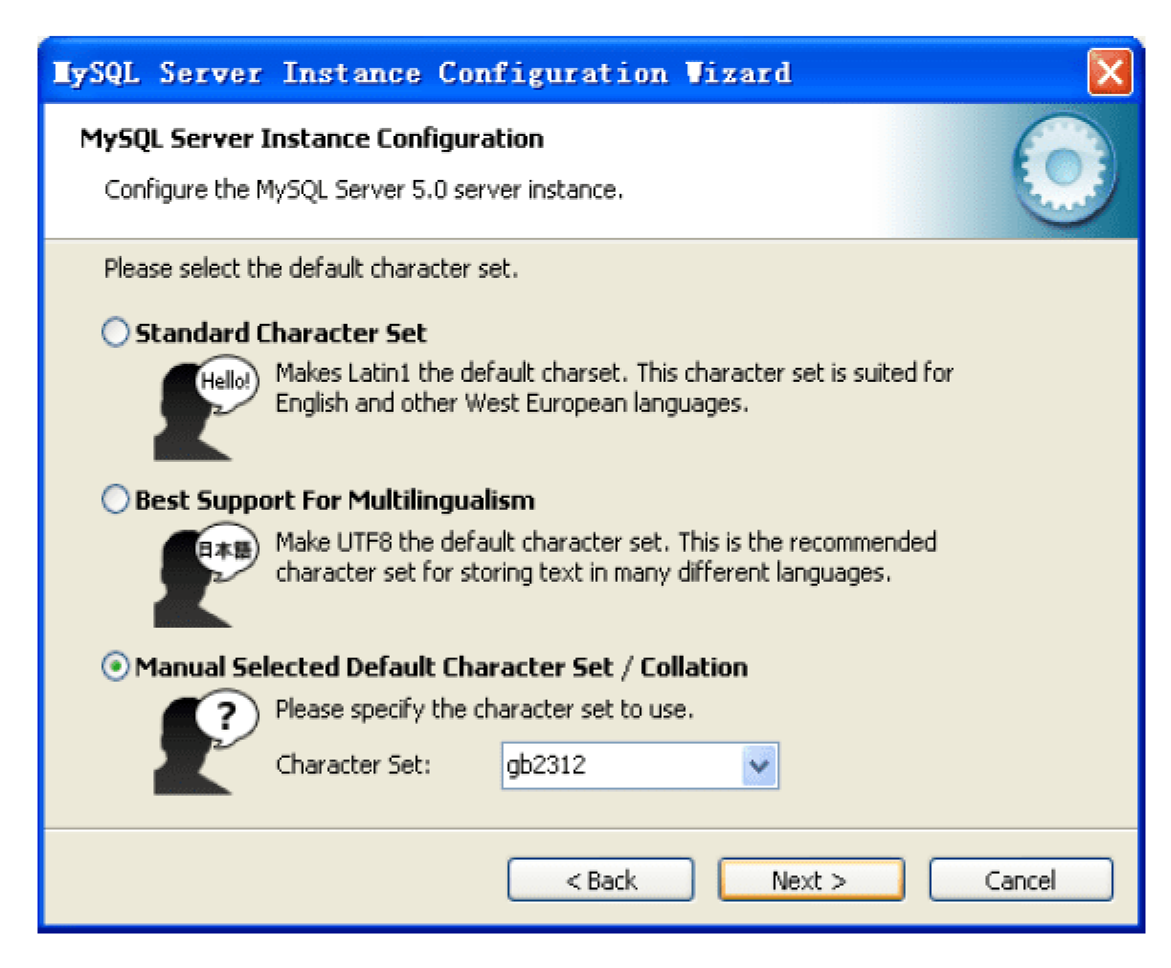


图 8-7 字符集设置

提 示

字符集的概念比较复杂，这里就不再详述。事实上一般来说采用什么样的字符集对 MySQL 影响不大，只有在不同 MySQL 之间导入/导出数据时才考虑字符集是否一致的问题，否则容易导致乱码。这里不妨选择第三个选项，并把“gb2312”即简体中文设置为字符集。

（8）继续单击 Next 按钮，出现 Windows 选项窗口，如图 8-8 所示。在这里可以选择是否将 MySQL 安装为 Windows 的服务。这里可以选择是，这样可以在机器启动时自动启动 MySQL 数据库服务。另外下面的“Include Bin Directory in Windows PATH”也建议选中。本选项的意思是将 MySQL 的 Bin 目录加入到 Windows 的环境变量中。这样做可以在命令行下直接运行 MySQL 命令，而无须先切换到 MySQL 的安装目录的 Bin 目录下（MySQL 的命令存放在安装目录下的 Bin 子目录下，如 C:\mysql5\bin）。



图 8-8 设置 windows 选项

（9）继续单击 Next 按钮，出现安全选项窗口。在这个窗口中可以设置 MySQL 数据库超级管理员的密码。MySQL 安装完毕之后默认生成一个用户名为 root 的超级管理员用户，密码为空。这个用户拥有对数据库的完全控制权限。因此这个密码非常重要，一旦设置了就要务必牢记，一旦忘记很难找回。如果是在服务器上安装 MySQL，这个密码务必要设置，而且设置得越复杂越好。如果仅仅是在本地机器作为学习、测试之用，为方便，可以暂不设置密码。这里将密码设置为“1234”，单击 Next。这时设置步骤完成，出现执行配置窗口，单击 Excute 按钮，开始执行配置，稍等片刻即可配置成功，如图 8-9 所示。

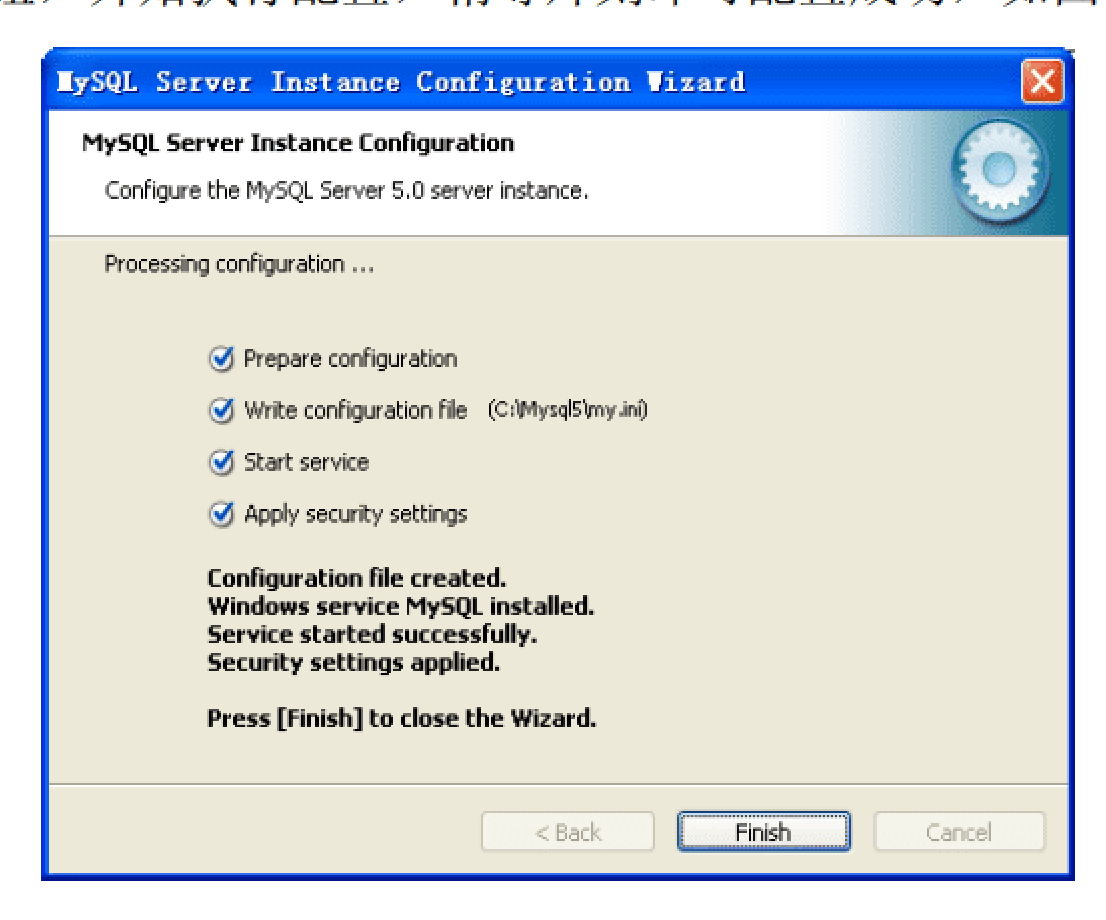


图 8-9　配置完成

这时所有配置工作都已顺利结束，单击 Finish 按钮结束配置程序。

8.2.3　测试 MySQL

如果安装配置成功，这时 MySQL 服务应该已经被启动。可以通过以下简单的测试，来验证 MySQL 安装是否成功：

（1）打开“开始”→“运行”，输入“cmd”，回车，会打开命令提示符窗口（也可以通过“开始”→“程序”→“附件”→“命令提示符”来打开）。

（2）在命令提示符下输入“mysql -u root –p”并回车，会出现“Enter password:”，输入密码“1234”回车。

（3）如果 MySQL 安装成功并已成功启动，将会出现如图 8-10 所示的登录成功的欢迎信息。

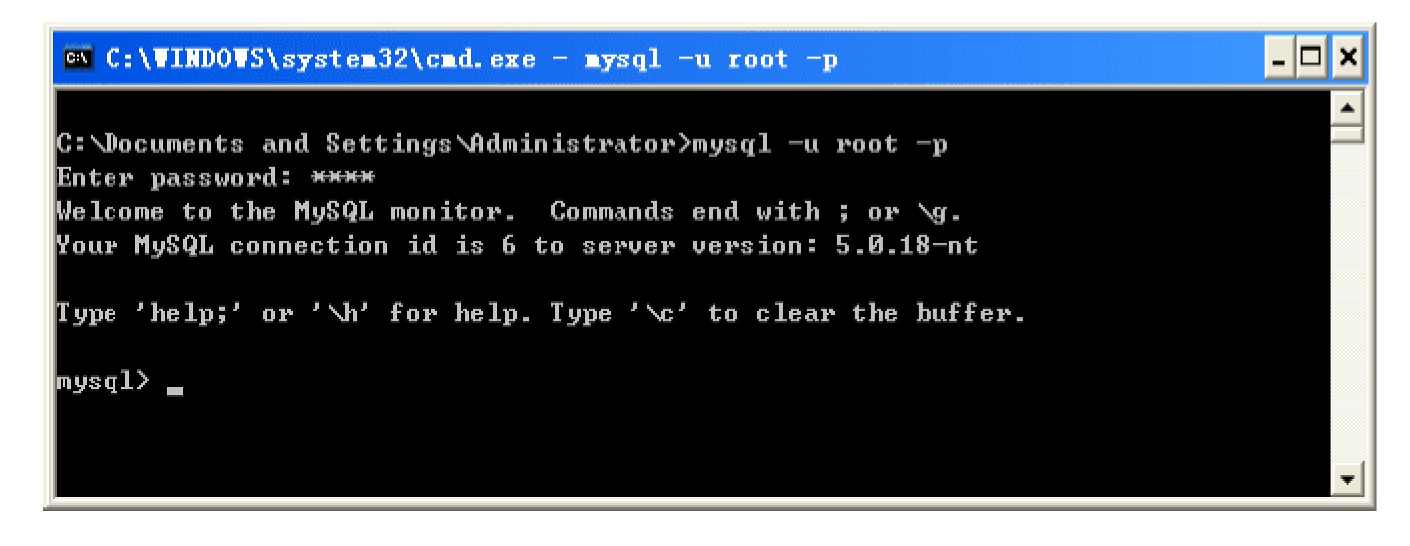


图 8-10　MySQL 登录成功的欢迎信息

根据软件环境不同，命令提示符的显示信息可能也会有不同。但只要输入密码后能够看到“Welcome to the MySQL monitor.”之类的提示，就说明 MySQL 登录成功。如果 root 用户没有设置密码，可以在出现“Enter Password:”之后直接打回车，以空密码进入。如果出现“MySQL 不是内部命令或外部命令，也不是可运行的文件或批处理文件”的提示，那说明在上面的配置步骤中，没有把 MySQL 的 Bin 目录加入到系统环境变量中，这时可以在命令提示符下切换到 MySQL 的安装目录的 Bin 目录，然后再输入“mysql -u root –p”命令来登录 MySQL。

8.3 MySQL 中的数据类型

为了对不同性质的数据进行区分，以提高数据查询和操作的效率，数据库系统将可存入的数据分为多种类型。如姓名、性别之类的信息为字符串型，年龄、价格、分数之类的信息为数字型，日期等为日期时间型。这就有了数据类型的概念。

数据类型是针对字段来说的。有的资料中称为“列类型”或“字段类型”。一个字段一旦设置为某种类型，那么这个字段中只能存入该类型的数据，不能写入非法数据。如“年龄”字段设置为整数型，那么数字 123 可以写入到这个字段中，字符串“ab”就无法写入到这个字段中。

就像编程语言一样，每种数据库都有自己支持的若干种数据类型。在数据库中建立表时，首先要考虑的就是这个表需要设置多少字段以及每个字段的数据类型。

MySQL 数据库中的数据类型分为三大类：数值类型、日期时间类型和字符串类型。各大类中包含的具体类型及其取值范围如表 8-2 所示。

表 8-2 MySQL 常用数据类型总表

大类	数据类型	取值范围或取值格式
数值型	TINYINT	有符号：-128~127 无符号：0~255
	SMALLINT	有符号：-32768~32767 无符号：0~65535
	MEDIUMINT	有符号：-8388608~8388607 无符号：0~16777215
	INT	有符号：-2147483648~2147483647 无符号：0~4294967295
	BIGINT	有符号：-9223372036854775808 ~9223372036854775807 无符号：0~18446744073709551615
日期时间型	DATETIME	0000-00-00 00:00:00
	DATE	0000-00-00
	TIMESTAMP	00000000000000
	TIME	00:00:00
	YEAR	0000

（续表）

大类	数据类型	取值范围或取值格式
字符串型	CHAR	0~255（字节，字符型）
	VARCHAR	0~65535（字节，字符型）
	BINARY	0~255（字节，二进制型）
	VARBINARY	0~65535（字节，二进制型）
	BLOB	无限大小（字节字符串）
	TEXT	无限大小（字符字符串）
	ENUM	枚举型，最多 65535 个元素
	SET	集合型，最多 64 个成员

在后面的章节中将陆续介绍其中一些最为常用的数据类型。

8.4　结构化查询语言（SQL）简介

8.4.1　结构化查询语言概述

结构化查询语言(Structured Query Language)，最早的是 IBM 的圣约瑟研究实验室为其关系数据库管理系统 SYSTEM R 开发的一种查询语言。SQL 结构简洁，功能强大，简单易学，所以自从 IBM 公司 1981 年推出以来，SQL 得到了广泛的应用。如今无论是像 Oracle 、Sybase、SQL server 这些大型的数据库管理系统，还是像 Visual Foxporo,PowerBuilder 这些桌面数据库开发系统，都支持 SQL 语言作为查询语言，MySQL 同样也不例外。

Structured Query Language 包含 4 个部分：

- 数据查询语言：SELECT；
- 数据操纵语言：INSERT,UPDATE,DELETE；
- 数据定义语言：CREATE,ALTER,DROP；
- 数据控制语言：COMMIT WORK,ROLLBACK WORK。

SQL 可用于所有用户的数据库活动模型，包括系统管理员、数据库管理员、应用程序员、决策支持系统人员及许多其他类型的终端用户。基本的 SQL 命令只需很少时间就能学会，高级的命令通过学习也不难掌握。SQL 可以完成的功能包括：

- 查询数据；
- 在表中插入、修改和删除记录；
- 建立、修改和删除数据对象；
- 控制对数据和数据对象的存取；
- 保证数据库一致性和完整性。

以前的数据库管理系统为上述各类操作提供单独的语言，而 SQL 将全部任务统一在一

种语言中。而且由于所有主要的关系数据库管理系统都支持 SQL 语言，因此用 SQL 编写的程序在一般情况下都具有可移植性。

上面都是 SQL 的理论性介绍，下面就来看一些具体的 SQL 语句，看 SQL 是如何实现对数据的操作的。

8.4.2 常用 SQL 语句的使用

1. CREATE 语句和 SHOW 语句

CREATE 语句可以用来创建新的数据库和表。SHOW 语句用来显示当前所有数据库或当前数据库下的所有表。下面看一个例子。

根据本章前面讲过的方法，打开命令提示符界面，输入用户名密码登录到 MySQL 控制台。登录到控制台后光标前面显示“mysql>”，在光标处可以直接输入 SQL 语句来操作数据库。现在来创建一个 student 的数据库。输入以下命令并回车：

```
mysql> CREATE  DATABASE  student;
```

SQL 语句可以用大写，也可以用小写，还可以大小写混合。本语句执行后会输出：

```
Query OK,1 row affected(0.08 sec)
```

这说明语句执行成功。一个名为 student 的数据库已经被创建成功。这时可以用以下命令来查看数据库是否已经被创建：

```
mysql> SHOW DATABASES;
```

输入命令后回车，会列出当前所有数据库。

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| student            |
| test               |
+--------------------+
4 rows in set(0.13 sec)
```

可以看到，student 数据库已经创建成功（information_schema、mysql、test 三个数据库均为 MySQL 安装时自动创建的原始数据库）。

然后，继续用 CREATE 语句在 student 数据库中创建一个表 info。这个表用来存储学生基本信息，一共有三个字段，分别是姓名(name)、性别(sex)、年龄(age)。这三个字段对应的数据类型分别为 CHAR、CHAR、TINYINT，长度限制在 20 字节、2 字节、2 字节以内。

在 student 数据库中创建表之前，需要首先执行这个语句，打开这个数据库：

```
mysql> USE student;
```

此语句执行后显示“Database changed”，表示数据库已经打开。

然后输入以下语句并回车：

```
mysql> CREATE TABLE info(name char(20),sex(2),age tinyint(2));
```

注意每条 SQL 语句输入完毕后最后要跟一个“;”，表示输入完成。否则不论怎样回车此语句都不会执行，直到遇到分号结尾。

语句执行完毕后显示“Query OK,0 rows affected(0.14 sec)”。表示语句执行成功。这时候表 info 应该已经创建成功。可以使用以下命令来查看 student 数据中现有的表。

```
mysql> SHOW TABLES;
```

执行后显示：

```
+--------------------+
| Tables_in_student |
+--------------------+
| info              |
+--------------------+
1 row in set(0.00 sec)
```

这时可以看到 info 表确实已经创建到了 student 数据库中。

2. DROP 语句

DROP 语句用来删除一个数据库或者一个表。如果是删除一个数据库，那么这个数据库下的所有表也将被删除。如以下语句：

```
mysql> DROP DATABSE D1;
```

将删除名为 D1 的数据库。

```
mysql> DROP TABLE tbl1;
```

将删除名为 tbl1 的表（删除前需要先打开数据库）。

3. INSERT 语句

INSERT 语句用来向表中插入新的数据记录，每次插入一条。比如要向刚才创建的 info 表中插入一条各字段值分别为“张三”、“男”、“20”的记录，可以这样写：

```
mysql> INSERT INTO info VALUES("张三","男",20);
```

执行过后显示“Query OK,1 row affected(0.08 sec)”，语句执行成功。

提 示

> 在插入数据时，字符串型值要用双引号或者单引号引起来，数值型不用引号（加引号就错了），而且提供的数据也必须按照表的字段顺序排列，不能颠倒。

下面将介绍如何从表中查询出插入的数据。但是在查询之前，先多执行几次 INSERT 语句来向表中多插入几条信息，这样可以更加形象地说明查询语句的作用。

4. SELECT 语句

SELECT 语句用来查询表中的数据。SELECT 语句是 SQL 中最复杂的语句之一。因为用 SELECT 语句可以实现极为复杂的查询功能，可以查询某个表中全部记录、部分满足条件的记录、全部字段、部分满足条件的字段等等，还可以同时从多个表中查询满足条件的记录，以及对查询结果进行排序等。

本书中仅介绍几种常用的 SELECT 语句，可以参考其他数据库专业书籍来学习其更加深入的用法。

① 查询全部记录全部字段

查询一个表中全部记录，可以用如下语句：

mysql> SELECT * FROM info;

这里“*”表示所有字段。info 为表名。程序执行后输出：

```
+------+------+------+
| name | sex  | age  |
+------+------+------+
| 张三 | 男   |  20 |
| 李四 | 男   |  18 |
| 王五 | 女   |  18 |
| 赵六 | 女   |  17 |
+------+------+------+
4 rows in set(0.02 sec)
```

可见刚才插入的四条数据全部被查询出来了。

② 查询全部记录部分字段

可以通过指定具体的字段和排序方式，来过滤掉不需要的字段。如要查询所有记录的姓名、年龄两个字段值，可以用如下语句：

```
mysql> SELECT name,age FROM info;
```

执行后输出

```
+------+------+
| name | age  |
+------+------+
| 张三 |  20 |
| 李四 |  18 |
| 王五 |  18 |
| 赵六 |  17 |
+------+------+
```

```
4 rows in set(0.00 sec)
```

③ **查询满足某个条件的记录**

通过 SELECT 语句的 WHERE 子句，可以查询某些满足指定条件的记录，这在查询中极为常用。如要查询所有年龄小于 19 的记录，可以用如下语句：

```
mysql> SELECT * FROM info WHERE age<19;
```

执行后输出：

```
+------+------+------+
| name | sex  | age  |
+------+------+------+
| 李四 | 男   |   18 |
| 王五 | 女   |   18 |
| 赵六 | 女   |   17 |
+------+------+------+
3 rows in set(0.01 sec)
```

当查询条件有多个时，可以使用 AND 关键字。如现在查询所有年龄小于 19 并且性别为女的记录，可以使用下列语句：

```
mysql> SELECT * FROM infor WHERE age<19 AND sex="女";
```

该语句执行后，将只输出满足条件的王五、赵六的两条记录。

④ **查询某些记录，并对结果进行排序**

使用 SELECT 语句的 ORDER BY 子句可以对查询结果进行排序。比如，要查询所有性别为“女”的记录，并且将结果按照年龄从小到大排序。

```
mysql> SELECT * FROM info WHERE sex="女" ORDER BY age ASC;
```

运行后输出结果如下：

```
+------+------+------+
| name | sex  | age  |
+------+------+------+
| 赵六 | 女   |   17 |
| 王五 | 女   |   18 |
+------+------+------+
2 rows in set(0.01 sec)
```

如果要将从小到大改为从大到小，则将命令中的“ASC”改为“DESC”即可。

5. UPDATE 语句

UPDATE 语句可以对表中现有的记录进行修改。

① **修改全部记录的某个字段的值**

例如我们要将 info 表中全部记录的年龄都修改成 25，可以写下面的语句：

```
mysql> UPDATE info SET age=25;
```

这时如果再用 SELECT 语句查询此表，会看到所有记录的 age 字段都变成了 25。

此外，还可以一次修改多个字段的值。比如除了要将所有记录的 age 字段修改成 25，还要将所有 sex 修改为“女”，可以用如下语句：

```
mysql> UPDATE info SET age=25,sex="女";
```

也就是说，多个字段之间用逗号隔开，可以一次修改多个字段的值。

② 修改满足某条件的记录

通过 WHERE 子句指定的条件，可以修改满足指定条件的记录的值。如要将姓名为“张三”的记录的年龄修改成 23，其他不变，则可以用如下语句：

```
mysql> UPDATE info SET age=23 WHERE name="张三";
```

执行之后再用 SELECT 语句查询此表，会发现“张三”的年龄为 23，其他的均为 25。

同理可以通过用逗号隔开的方法，修改满足指定条件的记录的多个字段。

6. DELETE 语句

DELETE 语句用来删除表中的记录。可以一次删除全部记录，也可以删除满足指定条件的记录。

① 删除表中的全部记录

如要删除表 info 中全部记录，可以用以下语句：

```
mysql>DELETE FROM info;
```

该语句执行后表 info 中的全部记录都会被删除。可以看出该命令是比较危险的，一旦不小心很容易造成误删，带来意想不到的损失。因此使用此命令时应尽量注意。

② 删除满足条件的记录

如果要删除表 info 中性别为“女”的记录，可以用如下命令：

```
mysql> DELETE FORM info WHERE sex="女";
```

读者可以自行验证此语句的效果。并尝试变换条件，来观察语句运行效果。

本节中扼要介绍了一些 Web 开发中最为常用的 SQL 语句。这些语句已经能够满足一般 Web 开发的需求。在后面的编程中会不断用到 SQL 语句，应注意多积累，多比较，多练习，掌握尽量多的 SQL 语句的使用方法，才能在以后的开发中得心应手。

8.5 MySQL 数据库管理工具 phpMyAdmin

在命令提示符下编写语句来进行数据库管理，虽然能够比较灵活地对数据进行操控，

但是却具有难度高、效率较低、容易出错等弊端。很多对命令提示符不熟悉的读者更是难以接受。实际上还有更加高效和便捷的方法来管理数据库，那就是借助现有的工具。phpMyAdmin 就是一个专门用来管理 MySQL 数据库的程序。这也是目前应用最为广泛的 MySQL 数据库管理工具。

8.5.1　phpMyAdmin 的安装

phpMyAdmin 不是一般的桌面应用软件，它是完全用 PHP 开发的一套程序，可以安装在安装了 PHP 和 MySQL 的机器上。通过浏览器来管理 MySQL 数据库。该程序功能强大，界面友好，是 PHP+MySQL 开发者十分青睐的工具软件。phpMyadmin 的压缩包可以从网络上下载，通过搜索引擎很容易就能找到其下载地址。在 http://www.mysql.cn 也可以下载到。该软件有诸多版本，在此以 2.8.0 版本为例来说明其使用方法。

（1）将下载到的压缩包解压缩，解压之后能得到一个包含了大量 PHP 程序的文件夹，此文件夹一般名为“phpMyAdmin-xx.xx.xx”。为了将来使用方便，可以将此文件夹重命名，不妨命名为 phpmyadmin。

（2）将此 phpmyadmin 文件夹复制到 IIS 或 APACHE 主目录下。这样就可以通过“http://localhost/phpmyadmin/”来运行此程序。

（3）打开 phpmyadmin/scripts/文件夹，找到 config.default.php，将其修改为 config.inc.php，拷贝到 phpmyadmin/目录下，也就是拷贝到 phpmyadmin 根目录下。然后打开此文件，找到“$cfg['Servers'][$i]['password']=”项，将其值设置为 MySQL 的超级用户密码，如图 8-11 所示。

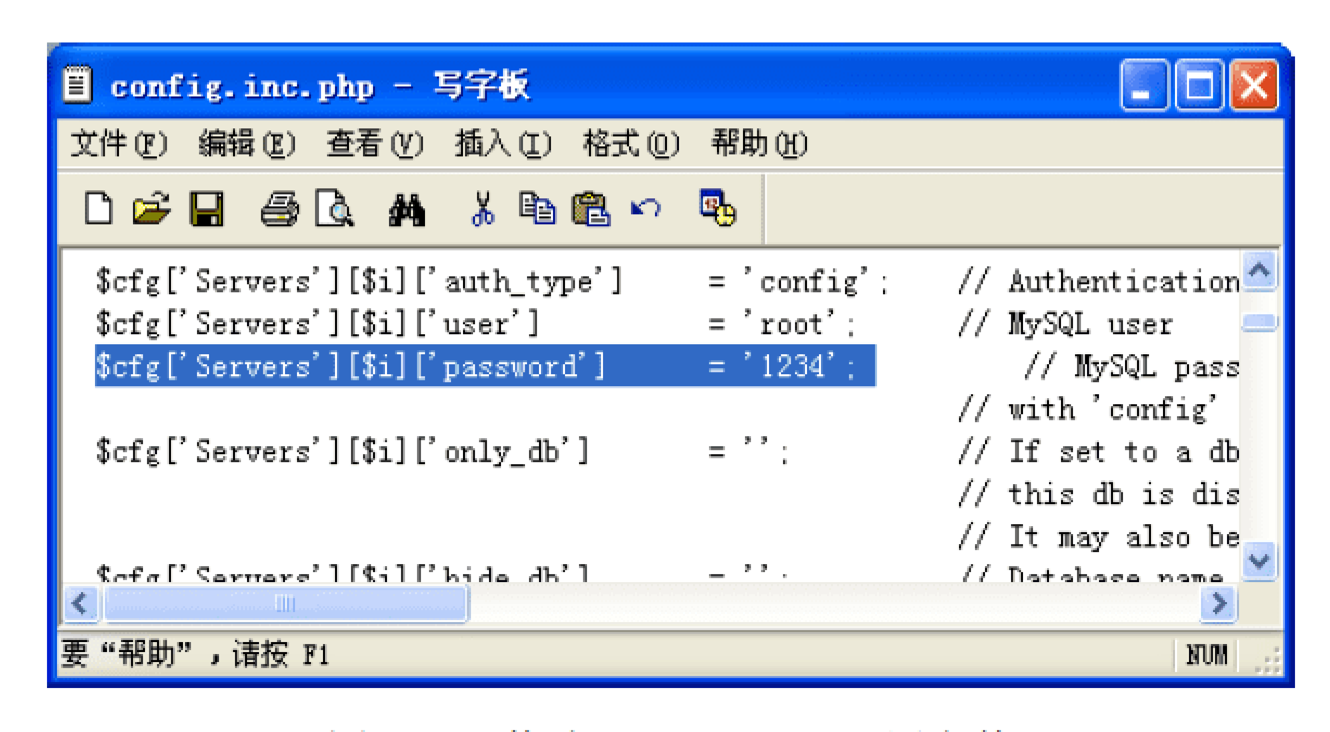


图 8-11　修改 phpmyadmin 配置文件

提 示

> 如果超级用户用户名不是 root，还需要将此文件中的“$cfg['Servers'][$i]['user']”修改为超级用户的用户名。

（4）修改完毕之后，保存文件。这时在浏览器中输入 http://localhost/phpmyadmin/index.php，就打开了 phpMyAdmin 的管理界面，如图 8-12 所示。

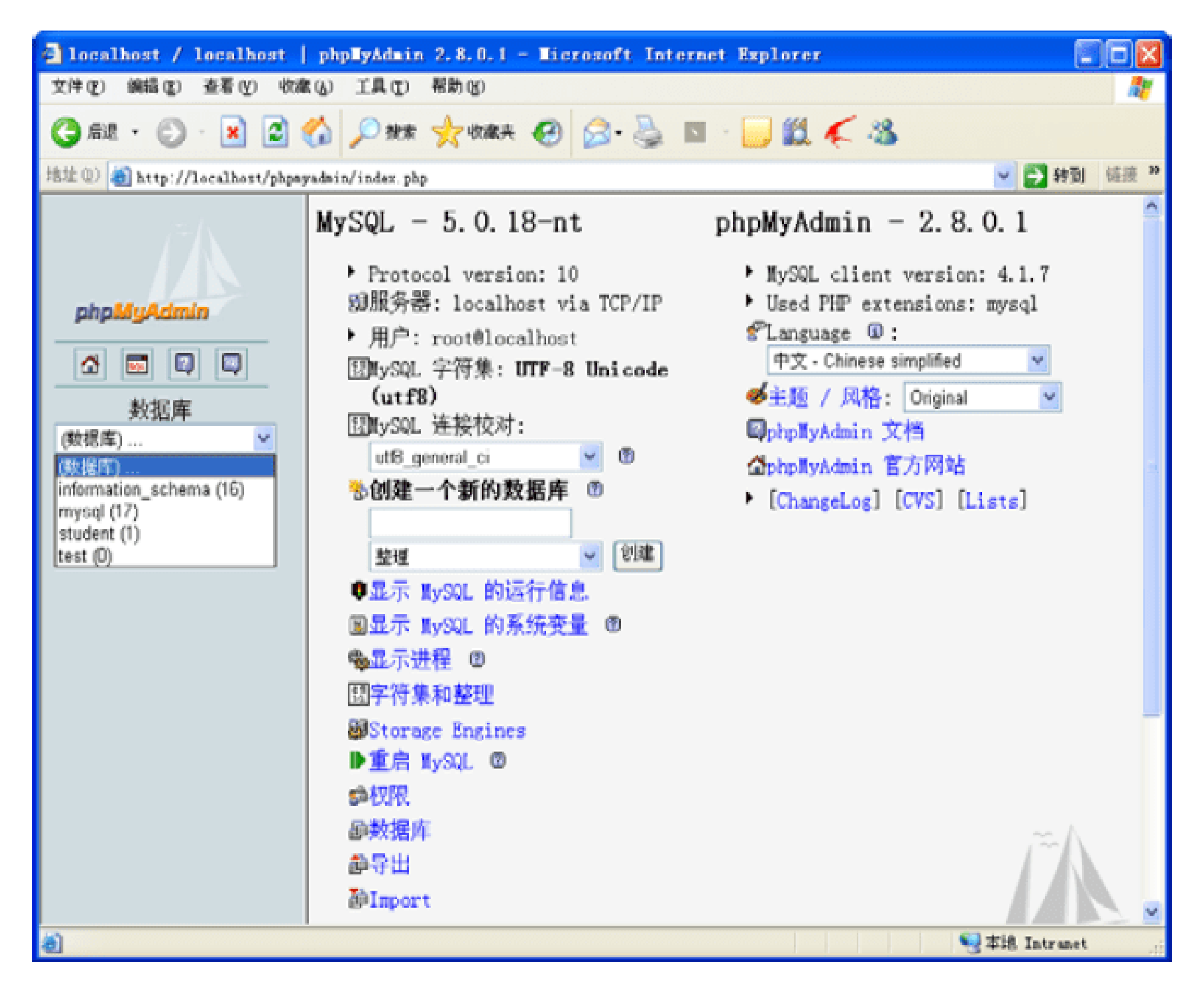


图 8-12 phpMyAdmin 运行界面

8.5.2 phpMyAdmin 的使用

在 phpMyAdmin 中，可以创建数据库、选择数据库、选择表信息、编辑数信息，下面介绍具体操作。

1. 创建新数据库

在管理界面右侧的“创建一个新数据库”下的表单中输入数据库的名字，单击“创建”按钮，即可创建一个新数据库。

2. 选择数据库

在左侧的下拉菜单中，单击选择一个数据库，那么该数据库就被选中了，下面将列出此数据库中的所有表，右侧将列出数据库中表的信息，如图 8-13 所示。

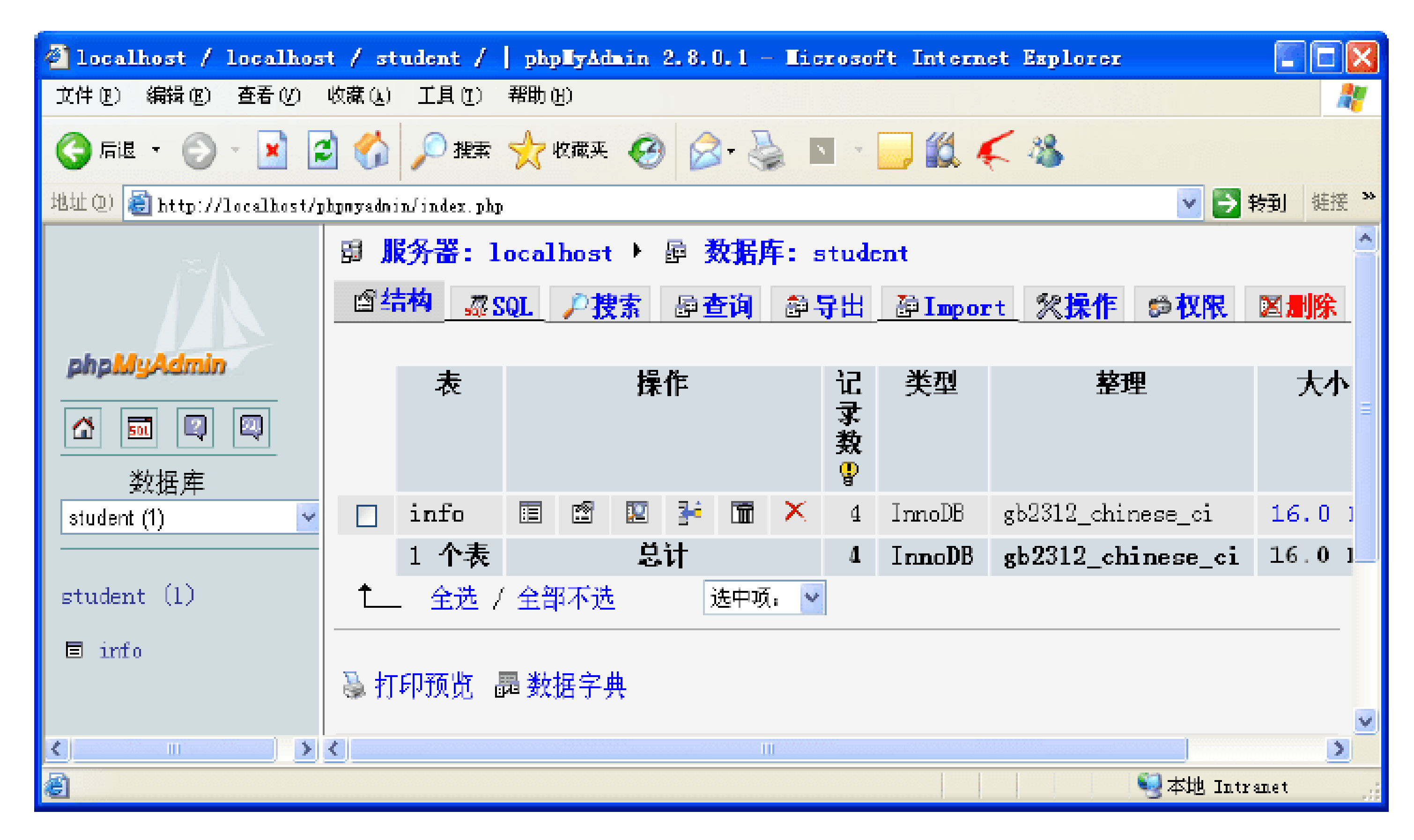


图 8-13 选择数据库

3. 选择并浏览表信息

单击左侧的表名，右侧将显示此表的详细字段信息，如图 8-14 所示。

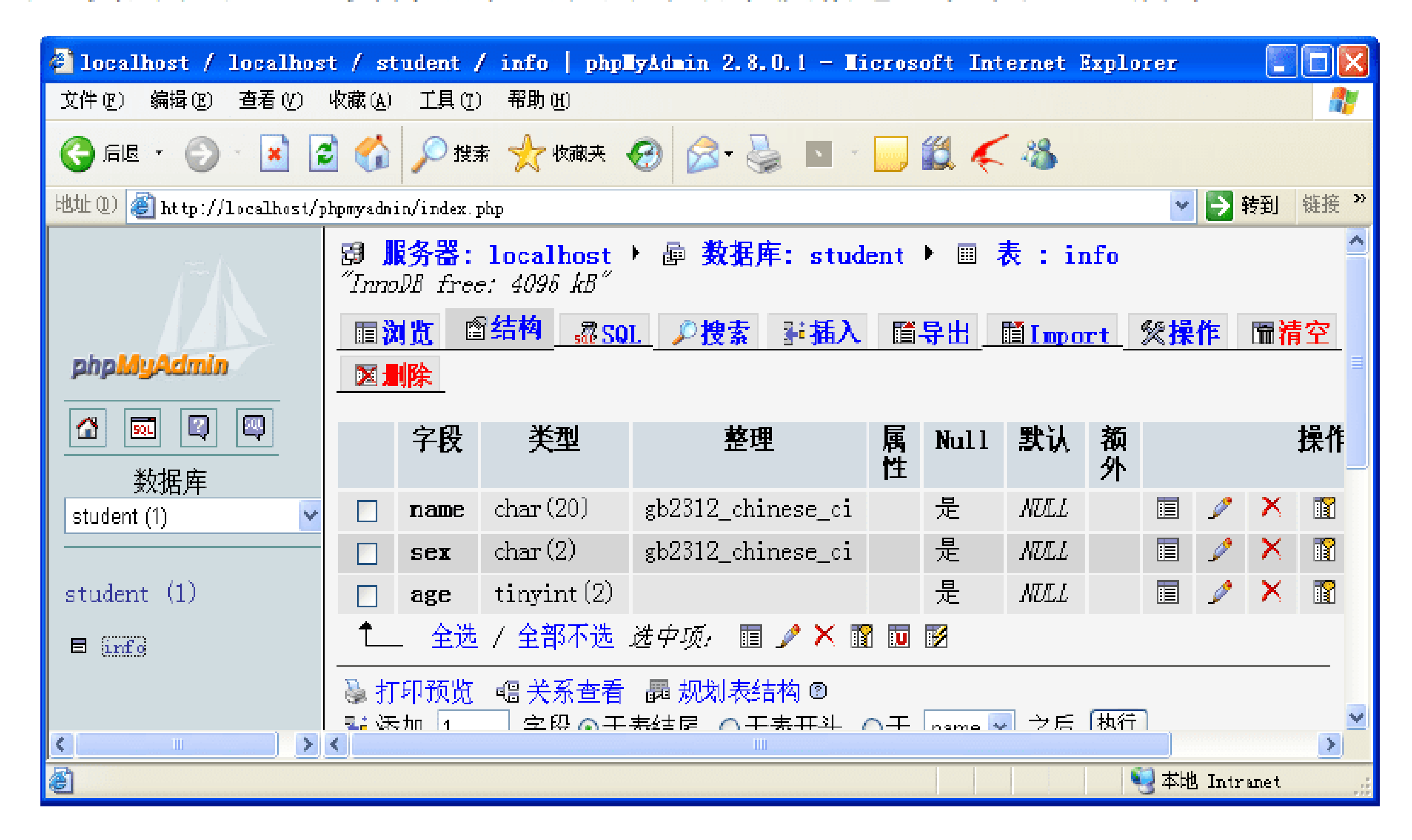


图 8-14　浏览表信息

这时单击右侧上方的按钮，可以进行浏览表内记录、表的结构、执行 SQL 语句、搜索、插入数据、修改数据等等操作。

4. 浏览及编辑数据

单击表上方的“浏览”之后，会打开该表的记录浏览。在此视图中可以对各条记录进行浏览、修改、删除等操作，如图 8-15 所示。



图 8-15　记录的浏览和编辑

phpMyAdmin 是一款功能强大的软件，用它可以很容易地对 MySQL 数据库进行各种管理。本书限于篇幅，就不再详细讲解其使用方法，只是将此工具介绍给广大读者，请各位读者自行安装、试用，掌握其常用功能，这对于提高数据库管理效率将起到很大的帮助作用。

8.6 案例 8 安装配置 MySQL 并建立一个用户信息表

8.6.1 案例分析

本案例的第一部分是安装并配置 MySQL。只要根据本章所讲的内容，按部就班，逐步操作，就可以轻松完成。此案例的目的就是让你亲自动手，熟练掌握 MySQL 的安装与配置步骤。

本案例的第二部分是建立一个用户信息表。可以用 SQL 在命令提示符下建立，也可以用 phpMyAdmin 来建立。这里使用 phpMyAdmin。

结合案例 7 中用户注册与登录的案例，很容易分析出，这个用户信息表只需要取代文本文件的功能即可。用户信息只有登录名、登录密码和头像三个字段。这样就可以设计一个三个字段的表，分别用来存储登录名、登录密码和头像地址（要注意我们这里并不是把图片直接存入表中，而是把图片的名字和路径存入表中）。

另外，由于本案例不要求新建一个数据库，因此不妨把表建立在前面已经建成的 student 数据库中。

8.6.2 实施步骤

（1）打开 phpMyAdmin，选择左侧的“student”数据库，右侧出现数据库属性。找到“在数据库 student 中创建一个新表”，然后在“名字”中输入表的名称，不妨为 userinfo，在“Number of fields”即“字段个数”中输入 3，如图 8-16 所示。

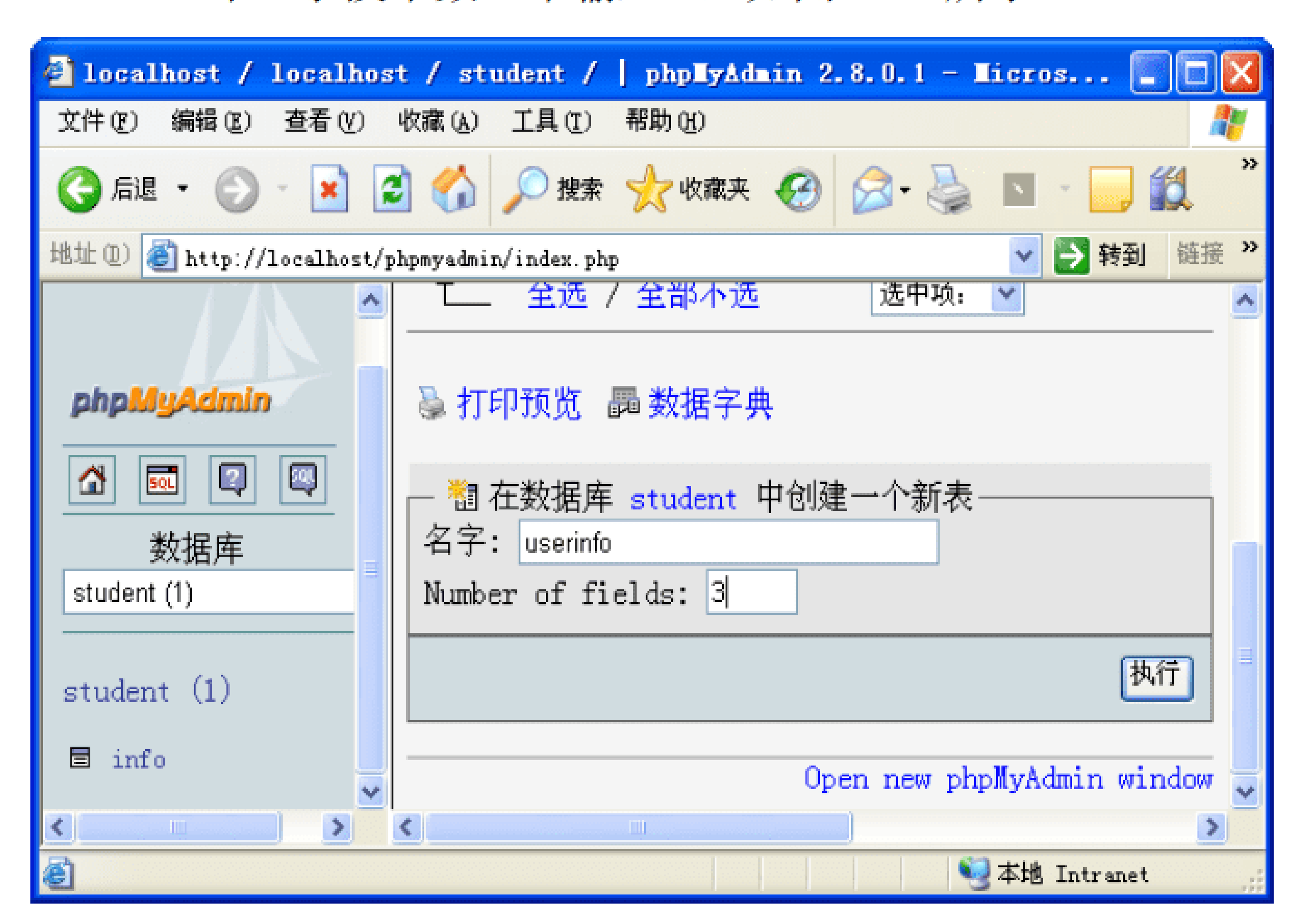


图 8-16 创建一个新表

（2）然后单击“执行”。此时会出现表字段设置界面，根据需求，将三个字段都设置为 varchar 类型，并为其指定最大字节数。其他属性可以根据需要进行设置。如图 8-17 所示。

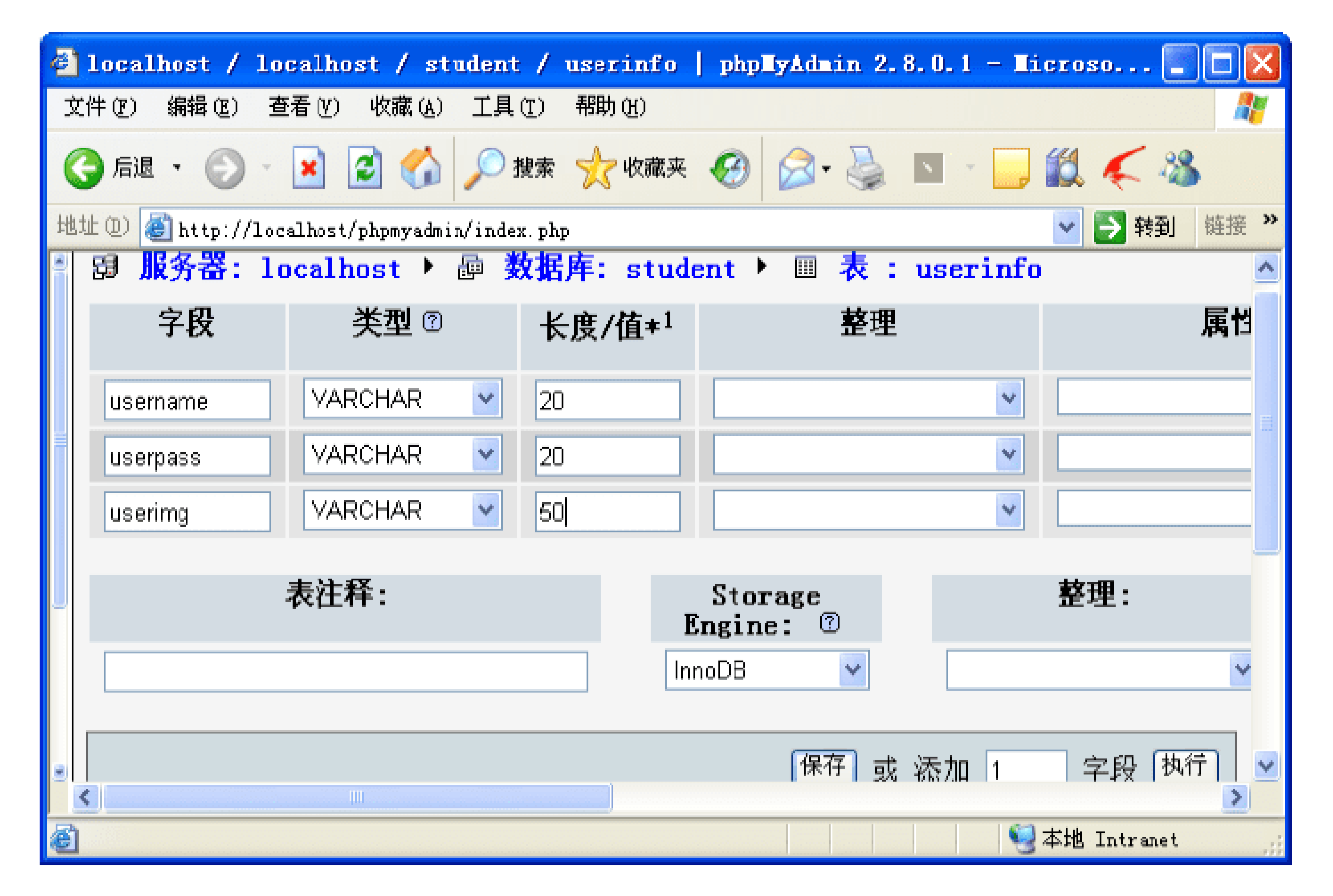


图 8-17　字段属性设置

（3）设置完成后，单击“保存”，此表即创建成功。这时再看左侧，已经有 info 和 userinfo 两个表了，如图 8-18 所示。

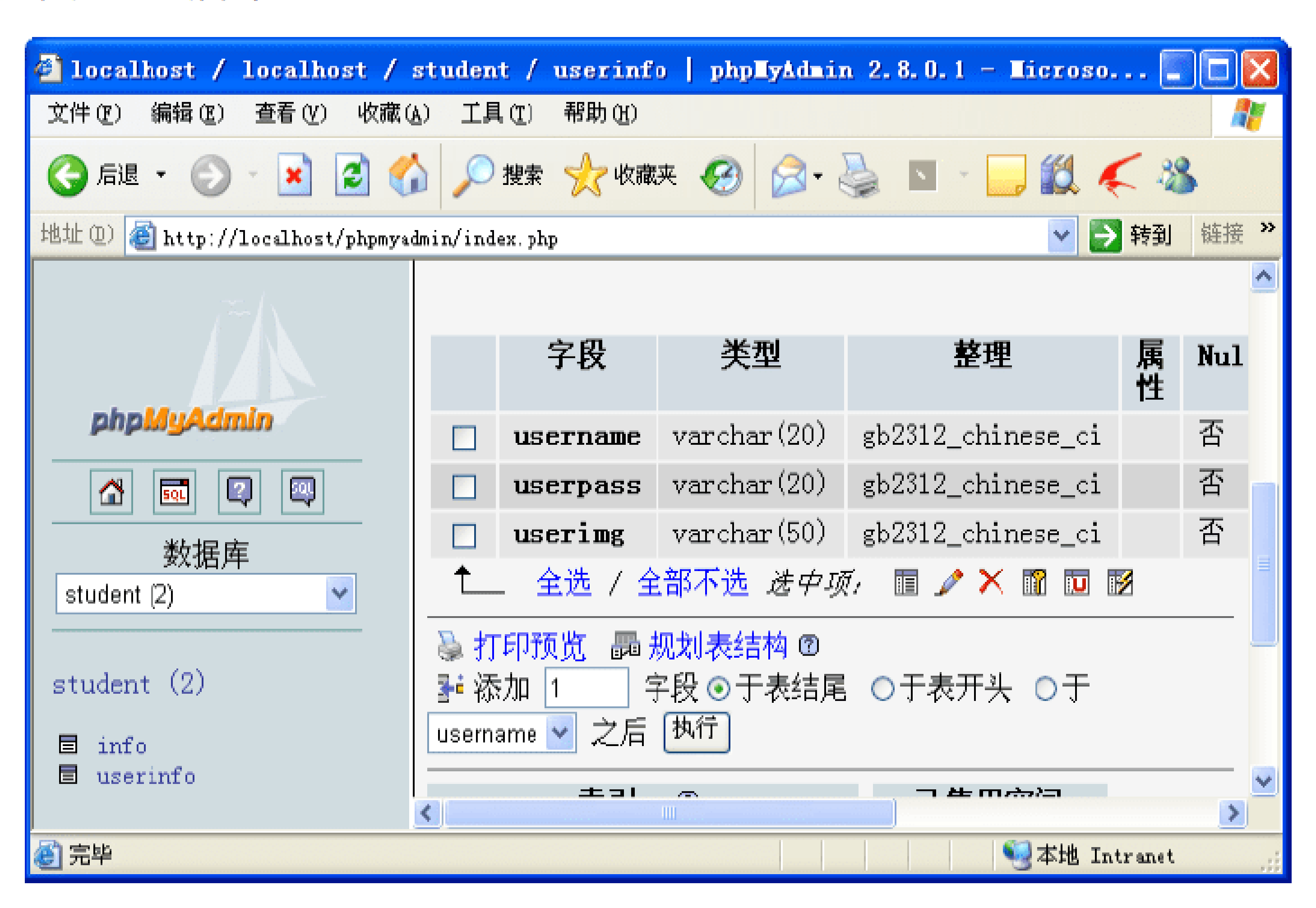


图 8-18　表创建成功

至此，本实例制作完成。

8.7　本 章 小 结

本章中主要介绍了 MySQL 数据库的安装和配置步骤，并介绍了 SQL 的基本情况以及常用的 SQL 语句的写法。另外还介绍了一款功能强大的 MySQL 数据库管理工具 phpMyAdmin。通过本章的学习，读者应熟练掌握 MySQL 数据库的安装配置方法，能够熟练使用基本的 SQL 语句进行数据库操作。能够熟练使用 phpMyAdmin 进行数据库管理。

8.8 练习题

1. MySQL 安装过程中有哪些值得注意的问题？
2. 在 Web 开发中使用数据库有何优点？
3. 常用的 SQL 语句有哪些？如何对查询结果进行排序？
4. 分别用 SQL 语句和 phpMyAdmin 创建表 register，其各字段情况如表 8-3 所示。

表 8-3 表 register 的字段名与类型

字段名	类型
name	字符
age	数值
birth	日期

CHAPTER

9

PHP+MySQL 数据库编程

本章重点

- PHP 操作 MySQL 数据库的流程
- 常用 PHP 数据库函数
- 数据的插入、删除和修改
- PHP 数据分页的实现

PHP + MySQL

9.1 PHP 操作 MySQL 数据库的流程

PHP 是一门 Web 编程语言，MySQL 是一款网络数据库系统。这二者是目前 Web 开发中最黄金的组合之一。那么 PHP 是如何操作 MySQL 数据库的呢？只有对 PHP 操作 MySQL 数据库的流程有一个基本了解，才能更加准确的理解 PHP 数据库编程的思路，为后面的学习打下基础。

从根本上来说，PHP 是通过预先写好的一些函数来与 MySQL 数据库进行通信，向数据库发送指令、接收返回数据等都是通过函数来完成。如图 9-1 所示给出了一个普通 PHP 程序与 MySQL 进行通信的基本原理示意图。

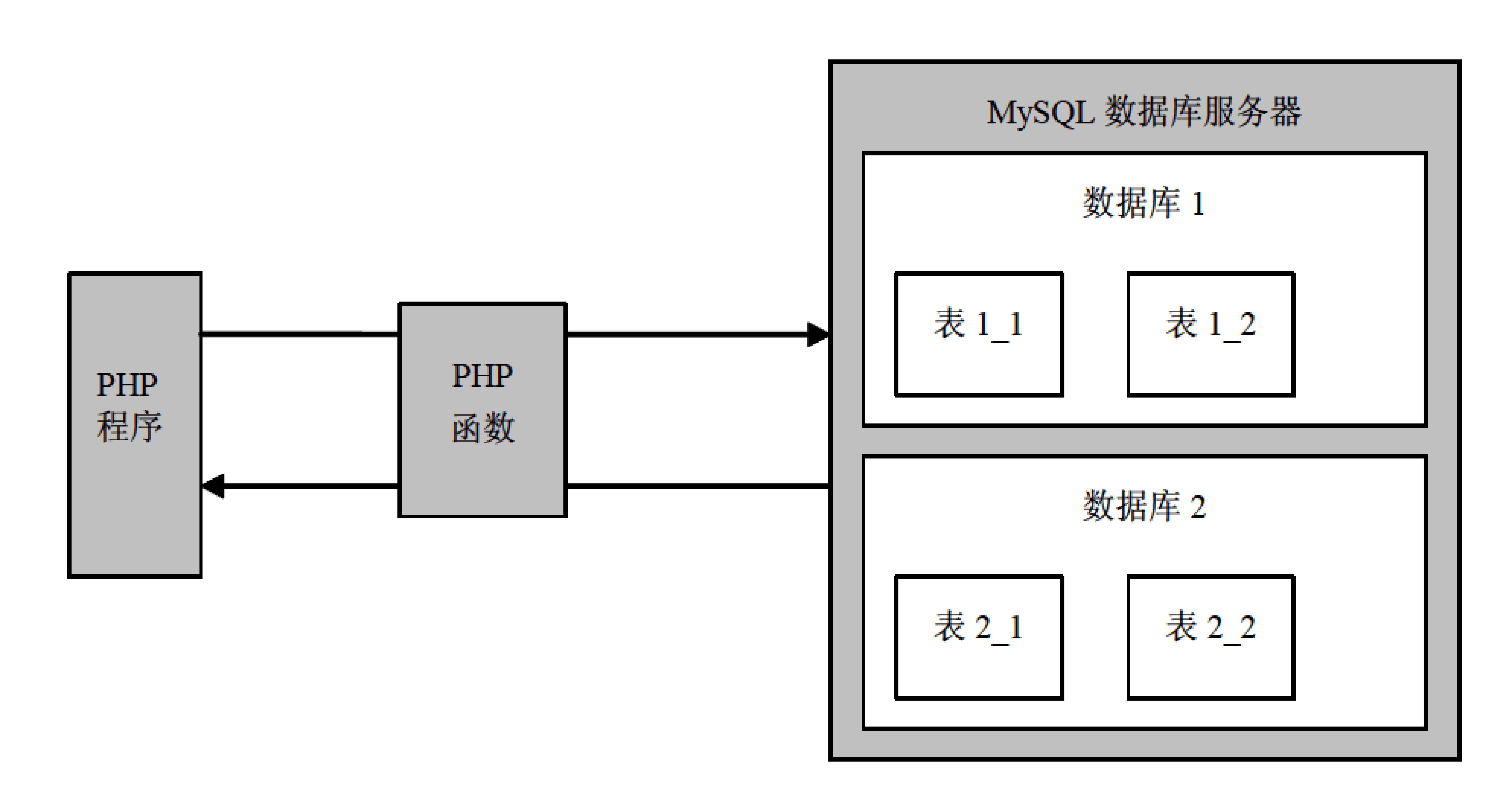


图 9-1　PHP 程序与 MySQL 数据库通信原理示意图

上图描述 PHP 程序连接到 MySQL 数据库服务器的基本原理：PHP 通过调用自身的专门用来处理 MySQL 数据库连接的函数来实现与 MySQL 通信。而且，PHP 并不是直接操作数据库中的数据，而是把要执行的操作以 SQL 语句的形式发送给 MySQL 数据库服务器，由 MySQL 数据库服务器执行这些指令，并将结果返回给 PHP 程序。MySQL 数据库服务器可以比作一个数据“管家”。其他程序需要这些数据时，只需要向“管家”提出请求，“管家”就会根据要求进行相关的操作或返回相应的数据。

如图 9-2 所示展示了从 PHP 代码到最终取得数据的流程。

明白了 PHP 操作 MySQL 的流程，就很容易掌握 PHP 操作 MySQL 的相关函数。因为几乎以上每一个步骤，都有相应的函数与之对应。开发 PHP 数据库程序时，只需要按照流程调用相关函数，数据库操作便可轻松实现。

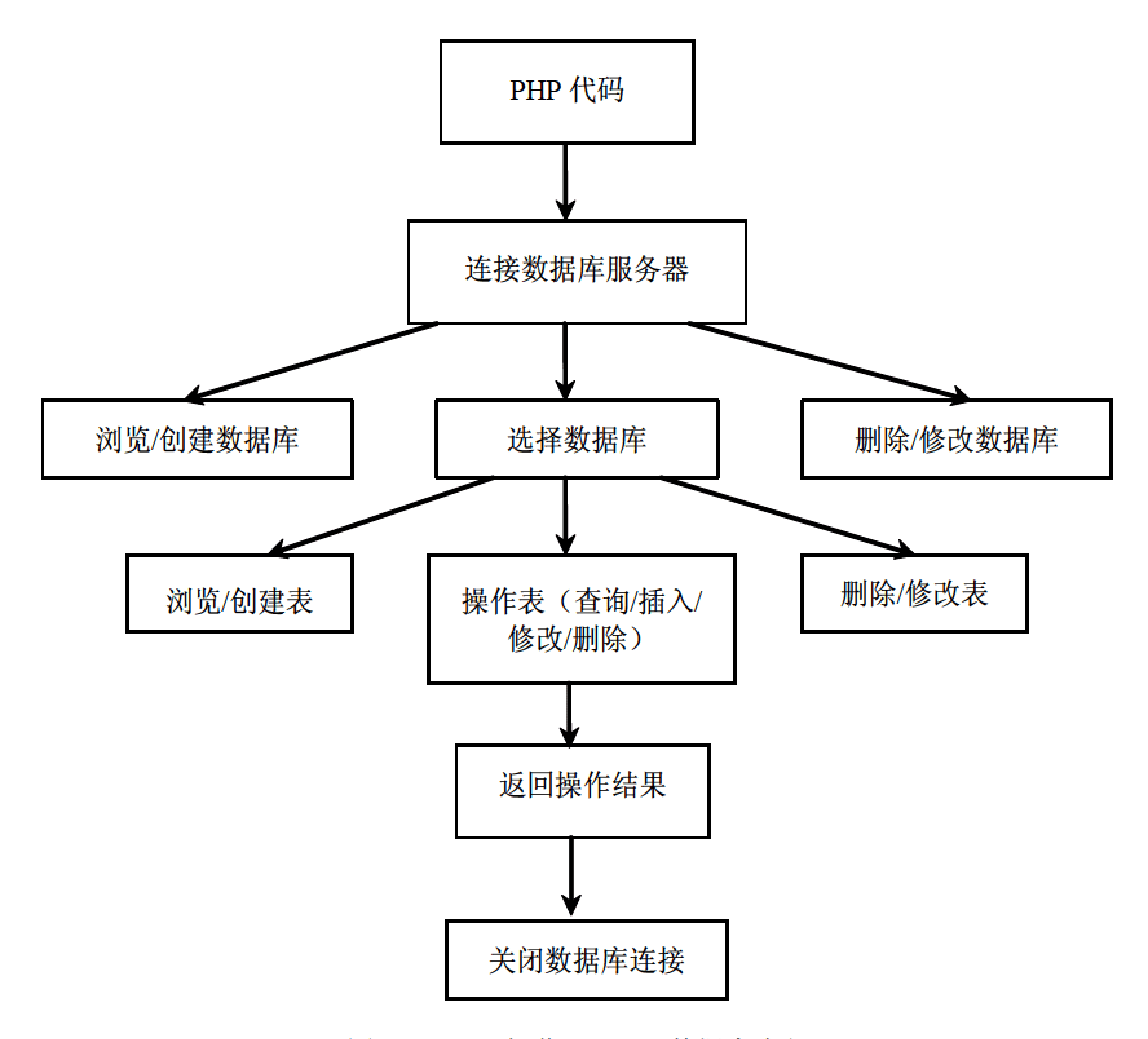


图 9-2　PHP 操作 MySQL 数据库流程

9.2　连接数据库前的准备工作

经过前面漫长的学习，现在终于要迈入 PHP+MySQL 数据库编程的大门了。不过在此之前，还有最后一点问题要注意，就是做好连接数据库前的准备工作，否则可能无法连接成功。

从 PHP 5 开始，不知道出于什么考虑，PHP 开发者放弃了对 MySQL 的默认支持，而是放到了扩展函数库中。因此要使用 MySQL 函数，需要首先开启 MySQL 函数库。

打开 php.ini，找到“;extensions=php_mysql.dll”，将此行前面的分号“;”去掉，保存之后重新启动 IIS/Apache。这时候仍然不能肯定这些函数已经被载入，可以通过 phpinfo()函数来查看一下。用 phpinfo()函数显示出 PHP 环境配置信息，然后查找一下里面有没有一个名为“MySQL”的项目。如果能找到，则说明 PHP 已经完全开启了对 MySQL 的支持，可以在程序中直接调用 MySQL 数据库了。如图 9-3 所示。

如果此时你的 phpinfo()程序显示不出 mysql 的信息，那么说明配置还没有成功。除了继续检查上一步修改是否正确以外，可以把 PHP 安装目录下的 libmysql.dll 这个库文件直接拷贝到系统的 system 目录或者 system32 目录下，拷贝之后再重新启动 IIS/APACHE，这时再次运行 phpinfo()程序，看是否出现了 mysql 信息。一般来说，拷贝 libmysql.dll 是最有把握的一种方法，正常情况下一定可以成功。

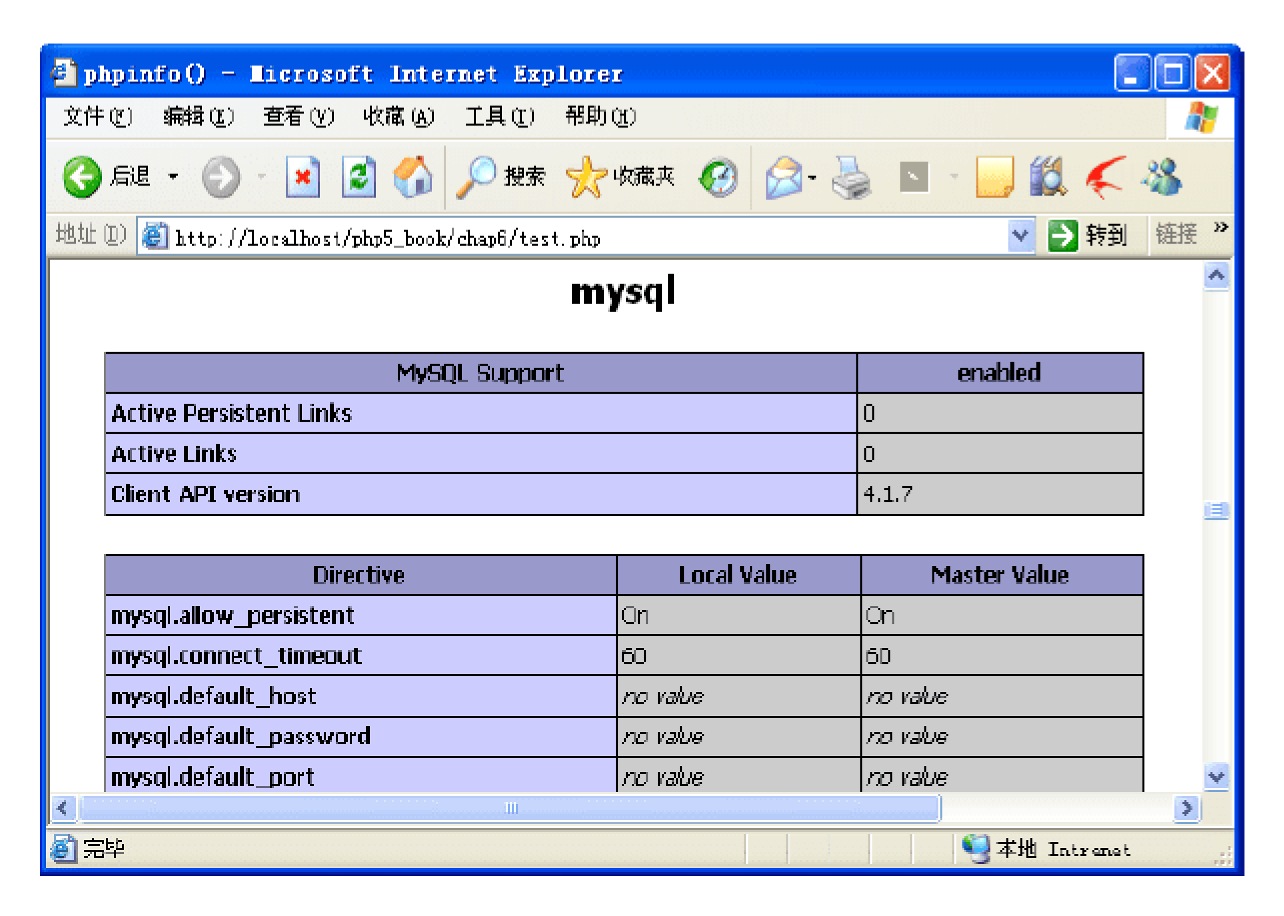


图 9-3　开启 MySQL 函数库

提 示

> 如果反复重复上述步骤，仍然不能成功开启 MySQL 函数库，那有可能是 PHP 安装包不完整，或者机器的软件环境有问题。可以通过正规渠道重新下载 PHP 安装包，重新配置 PHP，或者整理机器的软件环境来解决。

9.3　PHP 操作 MySQL 方法详解

9.3.1　PHP 操作 MySQL 常用函数

PHP 开发者提供了大量函数，这样可以方便的使用 PHP 连接到 MySQL 数据库，并对数据进行操作。学习 PHP+MySQL 数据库编程，首先要了解这些函数，明确具体的步骤，然后才能进入实质性开发阶段。

PHP 中可以用来操作 MySQL 数据库的函数如表 9-1 所示。

表 9-1　PHP 的 MySQL 函数一览

函数名	功能
mysql_affected_rows	取得前一次 MySQL 操作所影响的记录行数
mysql_change_user	改变活动连接中登录的用户
mysql_client_encoding	返回字符集的名称
mysql_close	关闭 MySQL 连接
mysql_connect	打开一个到 MySQL 服务器的连接
mysql_create_db	新建一个 MySQL 数据库
mysql_data_seek	移动内部结果的指针
mysql_db_name	取得结果数据

（续表）

函数名	功能
mysql_db_query	发送一条 MySQL 查询
mysql_drop_db	丢弃（删除）一个 MySQL 数据库
mysql_errno	返回上一个 MySQL 操作中的错误信息的数字编码
mysql_error	返回上一个 MySQL 操作产生的文本错误信息
mysql_fetch_array	从结果集中取得一行作为关联数组，或数字数组，或二者兼有
mysql_fetch_assoc	从结果集中取得一行作为关联数组
mysql_fetch_field	从结果集中取得列信息并作为对象返回
mysql_fetch_lengths	取得结果集中每个输出的长度
mysql_fetch_object	从结果集中取得一行作为对象
mysql_fetch_row	从结果集中取得一行作为枚举数组
mysql_field_flags	从结果中取得和指定字段关联的标记
mysql_field_len	返回指定字段的长度
mysql_field_name	取得结果中指定字段的字段名
mysql_field_seek	将结果集中的指针设定为制定的字段偏移量
mysql_field_table	取得指定字段所在的表名
mysql_field_type	取得结果集中指定字段的类型
mysql_free_result	释放结果内存
mysql_get_client_info	取得 MySQL 客户端信息
mysql_get_host_info	取得 MySQL 主机信息
mysql_get_proto_info	取得 MySQL 协议信息
mysql_get_server_info	取得 MySQL 服务器信息
mysql_info	取得最近一条查询的信息
mysql_insert_id	取得上一步 INSERT 操作产生的 ID
mysql_list_dbs	列出 MySQL 服务器中所有的数据库
mysql_list_fields	列出 MySQL 结果中的字段
mysql_list_processes	列出 MySQL 进程
mysql_list_tables	列出 MySQL 数据库中的表
mysql_num_fields	取得结果集中字段的数目
mysql_num_rows	取得结果集中行的数目
mysql_pconnect	打开一个到 MySQL 服务器的持久连接
mysql_ping	Ping 一个服务器连接，如果没有连接则重新连接
mysql_query	发送一条 MySQL 查询

（续表）

函数名	功能
mysql_result	取得结果数据
mysql_select_db	选择 MySQL 数据库
mysql_stat	取得当前系统状态
mysql_tablename	取得表名
mysql_thread_id	返回当前线程的 ID

这些函数中，最常用的有 mysql_connect()、mysql_select_db()、mysql_query()、mysql_fetch_array()、mysql_num_rows()、mysql_close()等。下面就着重介绍这几个函数的使用。

9.3.2 PHP 对 MySQL 数据库的操纵

PHP 对 MySQL 数据库的操纵分 4 大步，首先是建立与 MySQL 数据库的连接，然后是选择数据库，操纵数据，关闭与数据库的连接。

1. 用 mysql_connect()函数连接 MySQL 数据库

根据图 9-2 描述的流程，要在 PHP 中操纵 MySQL 数据库中的数据，第一步就是连接到数据库服务器。也就是建立一条 PHP 程序到 MySQL 数据库之间的通道。这样 PHP 才能通过这个通道来向 MySQL 服务器发送各种指令，并取得指令执行的结果，将这些结果应用于 PHP 程序中。mysql_connect()函数就是用来建立和 MySQL 数据库的连接。

mysql_connect()函数有 5 个参数，但是通常情况下只用到前三个参数。格式如下：

```
resource mysql_connect(string server,string username ,string password)
```

该函数返回类型为 resource 型，即资源型。3 个参数分别为 MySQL 服务器地址、MySQL 用户名、密码。这里的用户名可以用超级管理员的，也可以用用户表中存在的其他用户。如下面的语句将用超级管理员身份建立一个到本地服务器的连接：

```
$id=mysql_connect("localhost","root","1234");
```

其中“localhost”换成“127.0.0.1”或本地机器的实际 IP 地址，效果都是相同的。另外服务器地址后面可以指定 MySQL 服务的端口号，如果是采用默认的 3306 端口，则不必指定。如果采用了其他端口，则需要指定，如“127.0.0.1:88”表示 MySQL 服务于本地机器的 88 端口。用户名和密码均需指定（如密码为空，则直接用两个引号即可）。

将以上代码写在一个 PHP 程序中，写法如下：

```
<?php
$id=mysql_connect("localhost","root","1234");
echo $id;
?>
```

此程序运行之后，如果执行成功，则会输出一个资源型变量$id 的编号，类似于“Resource id #2”。如果执行失败，则有多种可能。如果出现下列提示：

```
Fatal error:Call to undefined function mysql_connect in [……]
```

则说明本地服务器的 MySQL 扩展库尚未被载入，因此 PHP 解释器无法识别 MySQL 函数。请参照本章第一节的内容进行重新设置。

如果出现下列提示：

```
Warning:mysql_connect()[function.mysql-connect]:Unknown MySQL server host [……]
```

则说明 MySQL 服务器地址错误，可能是输入有错误，或者服务器没有启动，或者段口号不对。这时可以检查函数的第一个参数是否提供正确，MySQL 是否已成功启动。

还有可能出现下列提示：

```
Warning:mysql_connect()[function.mysql-connect]:Access denied for user [……]
```

这说明用户名或密码有错误。或者本帐号没有在本 MySQL 服务器上登录的权限。

这里之所以如此详细地讲解该函数，就是因为是连接到 MySQL 数据库的第一步。因此必须保证此步骤成功，才能继续运行下面的函数。

2. 用 mysql_select_db()函数指定要操纵的数据库

连接到数据库以后，还不能直接操作某个表，因为表都存储在各个数据库当中，需要首先选择要操作的数据库，才能对这个数据库下面的表进行操作。mysql_select_db()函数就是用来指定要操作的数据库。前面的例子中们曾在 MySQL 中创建了一个 student 数据库，下面的代码将连接到数据库服务器，并把 student 数据库作为当前要操作的数据库。

程序清单　文档 9-1.php

```
<!--文件 9-1.php：连接到 MySQL 数据库服务器并选择数据库-->
1: <?php
2: $id=mysql_connect("localhost","root","1234");
3: $ok=mysql_select_db("student",$id);
4: if($ok){
5: echo "OK,选择数据库成功！";
6: }else{
7: echo "OH,选择数据库失败，请确认数据库是否存在。";
8: }
9: ?>
```

本程序中第 1 行建立了一个到本地 MySQL 数据库服务器的连接。第 2 行用 mysql_select_db()函数来指定要操作的数据库。函数第一个参数为数据库的名字，第二个参数为应用于哪个连接。第二个参数可以省略，省略时默认使用当前连接。一般来说，此参数可以直接省略。

mysql_select_db()函数返回一个布尔型值。如果执行成功返回 true，失败则返回 false。函数没有任何错误信息提示。因此即使提供的数据库名有错误或数据库不存在，本函数也不会报错。因此，返回结果存放在$ok 中，通过判断$ok 的值来判断是否执行成功。

3. 用 mysql_query()函数操纵数据库中的数据

连接到数据库服务器，并选择了要操作的数据库之后，下一步就是向服务器发送操作指令，也就是 SQL 语句。第 8 章中讲过的 SQL 语句，在这里可以派上用场了。现在来看一个例子，用 PHP 程序在 MySQL 中创建一个数据库 newdata，并在这个数据库中创建一个表 testtable，表的字段可以随意设置几个。

文档清单　文档 9-2.php

```
<!-- 文件 9-2.php：用 PHP 程序创建新数据库和表-->
<HTML>
<HEAD>
<TITLE>用 PHP 创建新数据库和表</TITLE>
</HEAD>
<BODY>
<?php
  $id=mysql_connect("localhost","root","1234");
  mysql_query("CREATE DATABASE newdata",$id);
  mysql_select_db("newdata",$id);
mysql_query("CREATE TABLE testtable(name varchar(10),age int(4))",$id);
  mysql_close($id)
?>
</BODY>
</HTML>
```

本程序执行完毕后不会有任何输出。但是如果用 phpMyAdmin 或者第 8 章讲过的命令提示符进入 MySQL 控制台，就会发现已经成功创建了一个名为 newdata 的数据库。打开此数据库，可以看到数据库中有一个 testtable 表。程序执行成功。

通过程序可以看出 mysql_query()函数的使用十分简单，只需要将一条 SQL 语句作为参数传递过去，即可执行此 SQL 语句。第二个参数$id 在一般情况下同样可以省略。

使用 mysql_query()函数可以向数据库服务器发送任何合法的 SQL 指令（前提是数据库要支持该指令）。文档 9-2.php 中只是测试了 CREATE 指令，实际上第 8 章讲的 INSERT、SELECT、UPDATE、DELETE 等指令同样可以用 mysql_query()来执行。下面就利用循环向服务器发送多次 INSERT 指令，向刚才创建的 testtable 表中插入多条记录。

程序清单　文档 9-3.php

```
<--文件 9-3.php：用 PHP 向表中插入数据-->
<HTML>
<HEAD>
<TITLE>用 PHP 向表中插入数据</TITLE>
</HEAD>
<BODY>
<?php
```

```
$id=mysql_connect("localhost","root","1234");
mysql_select_db("newdata",$id);
mysql_query("SET CHARACTER SET gb2312");
for($i=1;$i<6;$i++){
 $nl=20+$i;
 $xm="姓名".$i;
 $sql="INSERT INTO testtable VALUES('".$xm."',".$nl.")";
 $excu=mysql_query($sql,$id);
 if($excu){
  echo $sql;
  echo "第".$i."条数据插入成功! <br>";
 }else{
  echo "数据插入失败，错误信息：<br>";
  echo mysql_error();
 }
 }
mysql_close($id)
?>
</BODY>
</HTML>
```

本程序的运行结果如图 9-4 所示。

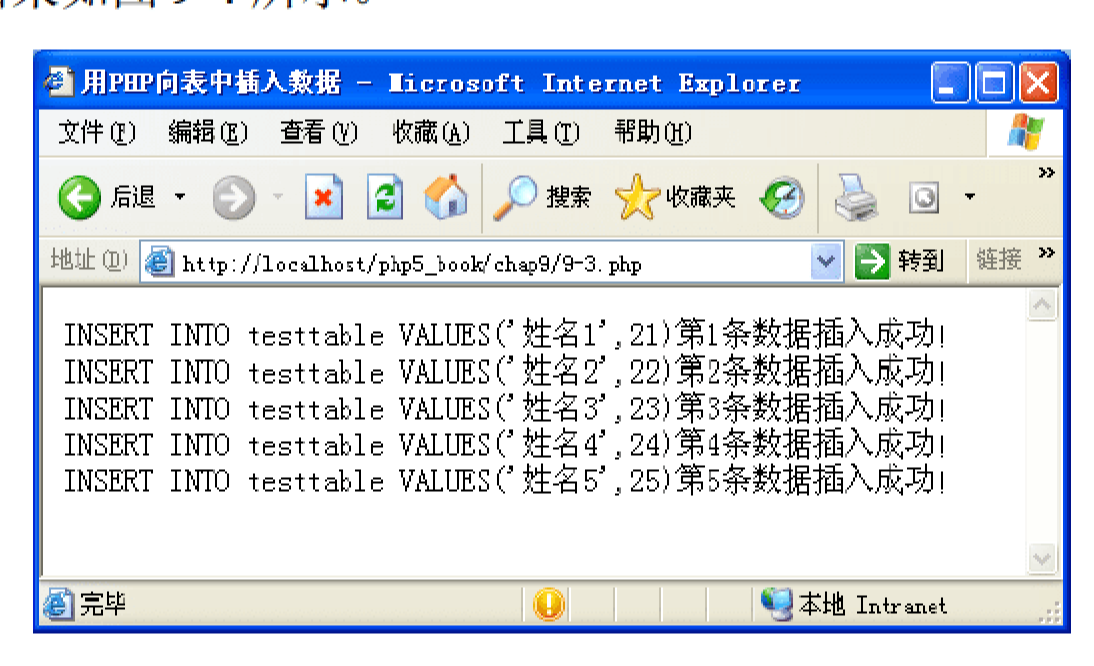


图 9-4　向表中插入数据

要验证 5 条信息是否都已确实插入到了数据库中，可以用命令提示符登录控制台，用“SELECT * FROM testtable”命令来浏览表中的所有数据，也可以用 phpMyAdmin 来浏览。控制台中显示的数据如下：

```
mysql> select * from testtable;
+--------+-------+
| name   | age   |
+--------+-------+
| 姓名 1 |   21  |
| 姓名 2 |   22  |
| 姓名 3 |   23  |
```

```
| 姓名 4 |   24 |
| 姓名 5 |   25 |
+-------+------+
5 rows in set(0.00 sec)
```

可见数据确实已经成功插入到数据库中。

本程序看起来行数较多，但是其结构却很简单。可以根据所学知识分析一下。在这里仅指出 3 点：

① 第 9 行中，用 mysql_query()函数首先向数据库发送了一条 SET CHARACTER 指令。这条指令好像与程序无关，为什么要用这条指令呢？原因是在本程序中向 name 字段中插入的值包含中文，如“姓名 1”、“姓名 2”，由于 PHP 向 MySQL 提交数据时默认采用 Latin1 字符集，如果不事先将字符集设置为“gb2312”，则很可能会出现字符集不匹配而导致插入失败（字符集置和采用的 MySQL 版本以及安装 MySQL 时的设置有关，有时候在程序中不需要设置字符集也能正常插入数据，因此读者应根据自己机器的环境特点决定是否有必要设置字符集）。

② 第 11~12 行，将要插入的两个字段的值放在两个变量中，然后在第 13 行构造一个 SQL 语句，第 14 行执行这个语句。这里之所以采用变量存放字段值和 SQL 语句，一是为了使程序更加易读，另外可以避免写成一条综合语句过于复杂而容易出错。要尤其注意的是第 13 行，在构造 SQL 语句（其实就是一个普通字符串）时，不同数据类型的变量用不同的引号等连接。其实这个问题应该是一个 PHP 基本语法的问题，很多初学者在这个地方容易犯迷糊，因此请多加揣摩，仔细体会。

③ 第 20 行首次使用 mysql_error()函数。这个函数可以返回上一次 MySQL 返回的错误信息。当程序出错时输出这些错误信息对于程序的调试很有帮助。可以试着故意写错 SQL 语句，或者故意发送一条非法指令到 MySQL 服务器，然后调用此函数查看返回的错误信息。

下面再来看一个从数据库中读取数据并用表格显示在网页上的例子。

还是用 mysql_query()函数，向数据库发送 SELECT 指令来查询数据。

程序清单　文档 9-4.php

```
<!--文件 9-4.php：用 PHP 从表中读出数据-->
1:<HTML>
2:<HEAD>
3:<TITLE>用 PHP 从表中读取数据</TITLE>
4:</HEAD>
5:<BODY>
6:<?php
7:  $id=mysql_connect("localhost","root","1234");
8:  mysql_select_db("newdata",$id);
9:  mysql_query("SET CHARACTER SET gb2312");
10: $query="SELECT * FROM testtable";
```

```
11: $result=mysql_query($query,$id);
12: echo "<table border=1 width=200><tr><td>姓名</td><td>年龄</td></tr>";
13: $datanum=mysql_num_rows($result);
14: echo "表 testtable 中共有".$datanum."条数据<br>";
15:  for($i=1;$i<=$datanum;$i++){
16:    $info=mysql_fetch_array($result,MySQL_ASSOC);
17:    echo "<tr><td>".$info['name']."</td>";
18:    echo "<td>".$info['age']."</td></tr>";
19:   }
20:  echo "</table>";
21:  mysql_close($id);
22:?>
23:</BODY>
24:</HTML>
```

本程序的输出结果如图 9-5 所示：

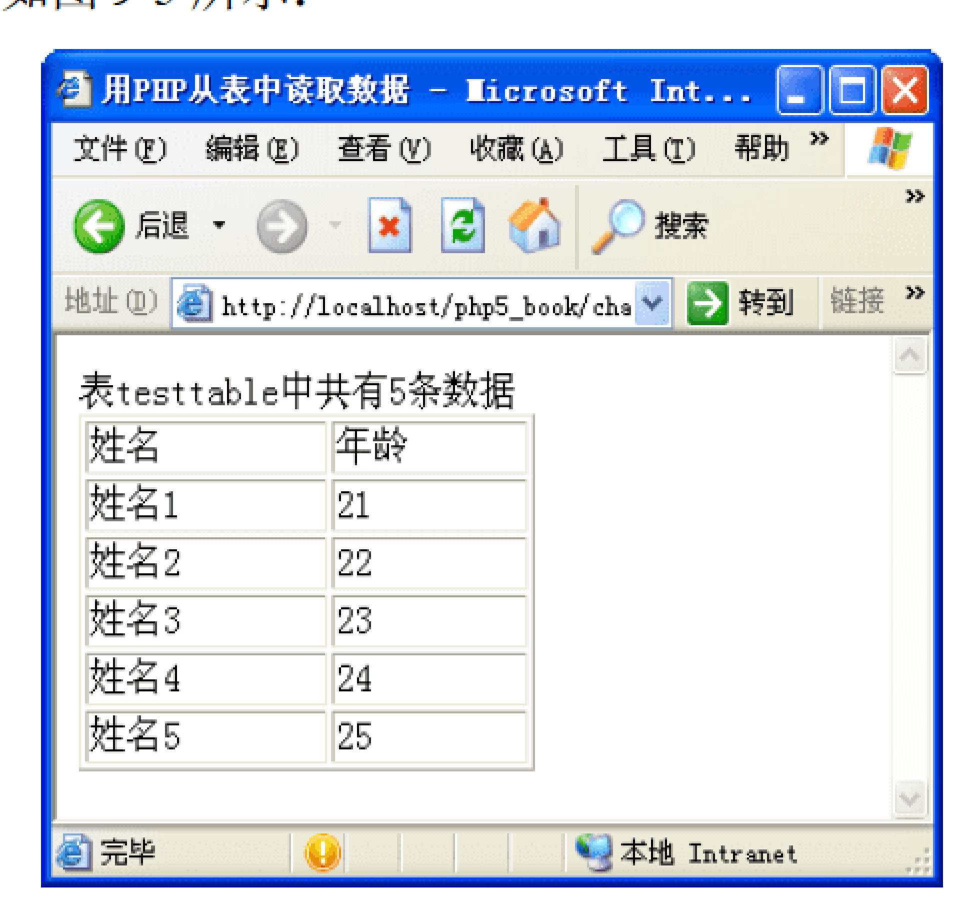


图 9-5　从数据表中读出数据

本程序中，代码中加入了一些 HTML 代码，用来产生一个表格。并用到了几个新的函数，下面就来一一介绍。

① 第 9 行的字符集设置语句作用与 9-3.php 中相似，防止读出的数据出现乱码。

② 第 11 行中，向服务器发送了一条 SELECT 指令。这条指令将返回所有满足条件的记录。注意返回的记录是一个资源类型，其内容是若干条记录的集合，可以成为一个记录集。不能直接用来输出，返回的数据存放在$result 中。

③ 第 13 行，用 mysql_num_rows()函数来统计一个记录集中记录的条数。注意此函数专用于统计 MySQL 查询结果记录集，不能用来统计其他数据类型的元素个数。

④ 第 16 行，这一行是关键。这一行用到了 mysql_fetch_array()函数。此函数是 PHP+MySQL 编程中最常用的函数之一。此函数的使用格式如下：

```
array mysql_fetch_array(resource result [,int result_type])
```

该函数的作用是，读取记录集 result 中的当前记录，将记录的各个字段的值存入一个数组中，并返回这个数组，然后将记录集指针移动到下一条记录。如果记录集已经到达末尾，则返回 false。

第二个参数 result_type 为可选，此参数用来设置返回的数组采用什么样的下标。有三个备选值：MySQL_ASSOC、MySQL_NUM、MySQL_BOTH。三个参数的含义如下：

- MySQL_ASSOC：返回的数组将以该记录的字段名称作为下标。如在本例中，要输出此数组中的“姓名”字段，可以用$info[‘name’]。这里$info 是数组名，“name”是存放姓名的字段名。
- MySQL_NUM：返回的数组以从 0 开始的数字为下标。在本例中，返回的每条记录只有两个字段，那么数组也就只有两个元素，分别用$info[0]、$info[1]来引用。
- MySQL_BOTH：返回的数组既可以用字段名为下标，也可以用数字为下标。在本例中，既可以用$info[0]来取得姓名，也可以用$info[‘name’]来取得。

可以自行修改程序，试验一下上述三个参数。

此外，PHP 中还有 mysql_fetch_row()、mysql_fetch_assoc()、mysql_fetch_object()等函数，这些函数的作用与用法都和 mysql_fetch_array()函数相似，可以参考 PHP 手册，了解一下这几个函数的使用。

用 mysql_query()函数，结合第 8 章中讲过的 SQL 语句，还可以轻松实现对表内数据的删除和修改操作。如下列代码：

```
<?php
$id=mysql_connect("localhost","root","1234");
mysql_select_db("newdata",$id);
mysql_query("DELETE FROM testtable",$id);
mysql_close($id);
?>
```

这段程序执行后，会删除表 testtable 中的全部数据。

此外，用 UPDATE 语句可以实现对表内数据的修改。在此不再举例，可以自行编写程序练习一下。

4. 用 mysql_close()函数半闭与数据库的连接

此函数用来关闭一个数据库连接。其使用格式如下;

```
bool mysql_close([resource link_identifier])
```

本函数只有一个可选参数 link_identifier。此参数表示要关闭的连接的 ID。也就是 mysql_connect()函数执行成功后返回的一个连接标记。参数为空时表示关闭当前连接。该函数返回一个布尔型结果。当关闭成功时返回 true，关闭失败则返回 false。

```
<?php
$id=mysql_connect("localhost","root","1234");
if(mysql_close($id)){
```

```
    echo "关闭数据库连接成功！";
}else{
    echo "关闭数据库连接失败！";
}
?>
```

上面的例子演示了 mysql_close()函数的使用。事实上，当一个 PHP 脚本（也就是一个 PHP 页面以及它的包含文件）执行结束时，在这个脚本中打开的 PHP 连接也会同时被关闭。因此一般情况下即使忘记了手动关闭也无关紧要，但是数据库使用完毕后关闭运行是一个很好的编程习惯。

9.3.3　PHP 中数据分页的实现

在 Web 开发中经常遇到的一个问题就是，对大量数据信息进行分页显示。比如一个留言板有数千条留言，如果这些留言全都显示在一个页面上，页面将变得很大难以浏览，有时过大的页面还会导致浏览器停止响应。PHP 中提供了非常简单方法，可以对数据进行分页。

为了更好的说明分页的作用，先向 testtable 表中插入足够大量的数据。修改 9-3.php，把控制插入条数的“for($i=1;$i<6;$i++)”修改为“for($i=1;$i<100;$i++)”，执行此程序之后，数据库中便会插入 99 条数据。这时再运行 9-4.php，会发现 99 条数据全部显示在一页内。

现在在 9-4.php 的基础上，为此程序增加分页功能。

程序清单　文档 9-5.php

```
<!-- 文件 9-5.php：用 PHP 实现数据分页显示功能-->
<HTML>
<HEAD>
<TITLE>用 PHP 实现数据分页</TITLE>
</HEAD>
<BODY>
<?php
  $id=mysql_connect("localhost","root","1234");
  mysql_select_db("newdata",$id);
  mysql_query("SET CHARACTER SET gb2312");

  $query="SELECT * FROM testtable";
  $result=mysql_query($query,$id);
  $totalnum=mysql_num_rows($result);
  $pagesize=15;
  $page=$_GET["page"];
  if($page==""){
    $page=1;
  }
```

```
$begin=($page-1)* $pagesize;
$totalpage=ceil($totalnum/$pagesize);
 echo "<table border=1 width=200><tr><td>姓名</td><td>年龄</td></tr>";
 $datanum=mysql_num_rows($result);
 echo "表 testtable 中共有".$totalnum."条数据。";
 echo "每页显示".$pagesize."条，共".$totalpage."页<br>";
 for($j=1;$j<=$totalpage;$j++){
  echo "<a href=9-5.php?page=".$j."><font color=black>[".$j."] </a>";
 }

$query="SELECT * FROM testtable limit $begin,$pagesize";
 $result=mysql_query($query,$id);
$datanum=mysql_num_rows($result);
 for($i=1;$i<=$datanum;$i++){
  $info=mysql_fetch_array($result,MySQL_ASSOC);
  echo "<tr><td>".$info['name']."</td>";
  echo "<td>".$info['age']."</td></tr>";
  }
 echo "</table>";
 mysql_close($id);
?>
</BODY>
</HTML>
```

程序运行结果如图 9-6 所示。

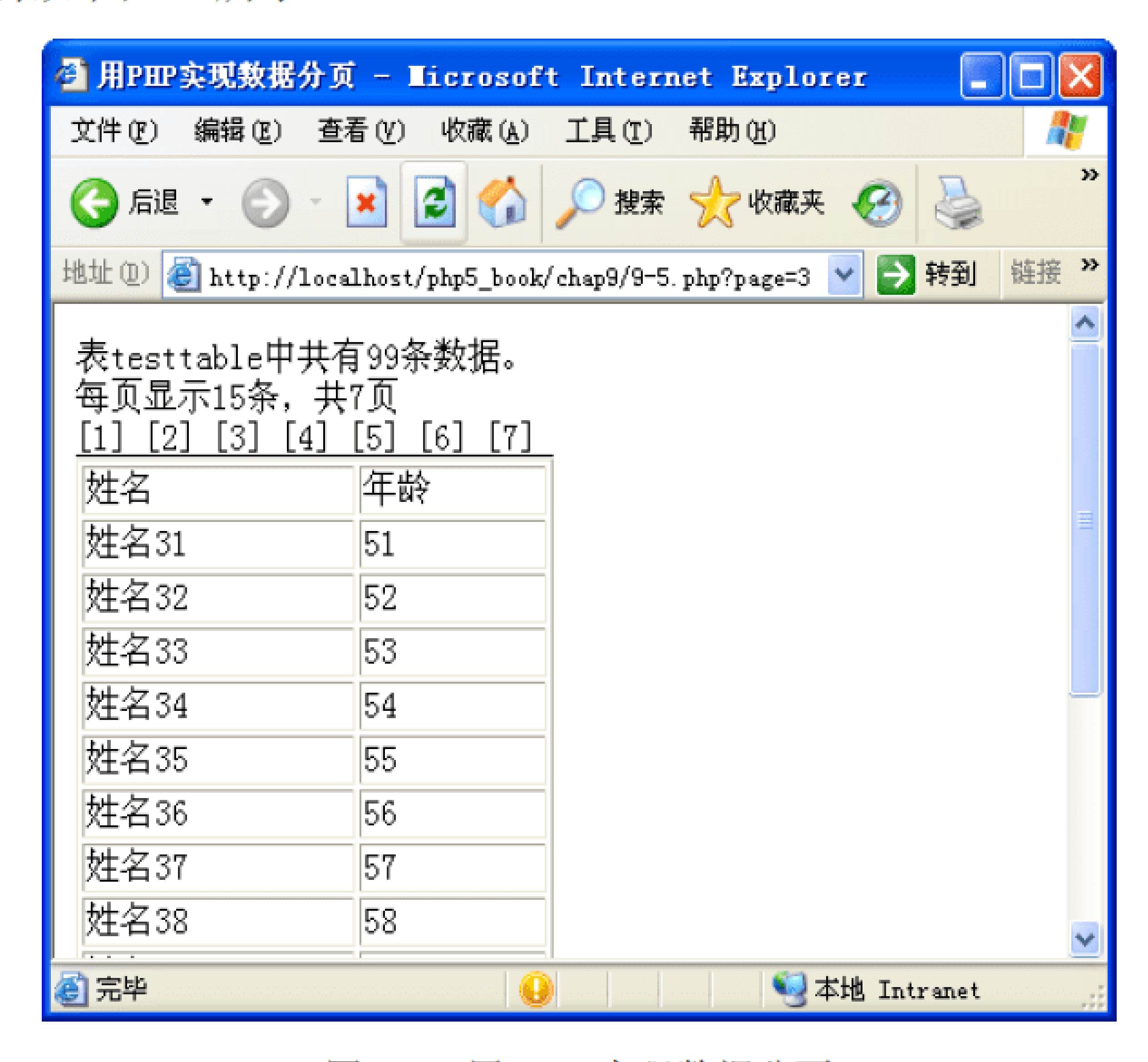


图 9-6 用 PHP 实现数据分页

单击不同的页码，对应的数据也会随之变化，简单的分页功能实现了。下面来解释一下本程序。

① 第 11~13 行，查询出要显示的全部数据的条数，放在$totalnum 中。

② 第 14 行定义了$pagesize 变量，将其值设置为 15，即每页显示 15 条数据。

③ 第 15~18 行判断是否单击了某一页。通过第 27 行可以看到，当用户单击某一个页码时，将会把一个页码的数字作为参数传递给本页面（用 GET 方法在 URL 中传递参数），第 15~18 行就是判断是否传递了这一参数，如果传递了，则说明用户选择了某一页，这时当前要显示的页面就是传递过来的参数值。如果参数为空或未传递，则说明本页面是直接被打开，用户还没有选择某个页面。这时就显示第一页，因此设置$page=1。

④ 第 19 行定义了变量$begin，这个变量在第 30 行的 SQL 语句中被用到。

MySQL 支持在 SELECT 语句中使用 LIMIT 子句，LIMIT 子句的使用方法为;

```
SELECT * FROM tbl WHERE … LIMIT [begin],[num]
```

此查询语句的含义为，查询出 tbl 表中满足指定条件的全部记录，然后只返回记录中从 begin 开始的 num 条数据。也就是说，使用了 LIMIT 子句后，MySQL 并不返回所有的满足条件的记录，而是只返回这些记录中从某条开始的若干条。如“LIMIT 10,5”的作用是取记录集中从第 10 条开始之后的 5 条，即第 11、12、13、14、15 条记录。注意不包括第 10 条本身。

这样，第 19 行的$begin 就不难理解了。记录截取起点的计算方法为：（要显示的页码-1）*每页显示的条数。如当前要显示第 3 页，每页 15 条，那截取起始位置就是(3-1)*15=30。也就是说，第三页从第 31 条数据开始显示，连续显示 15 条，直到第 45 条。这与实际情况是相符的。

⑤ 第 20 行根据记录总数和每页显示的条数计算出了总页数，计算方法很简单，就不再分析了。这里用到了第六章讲过的 ceil()函数来将计算结果进行进一取整。

⑥ 第 26~28 行，用循环输出了所有页码。每个页面都有一个超连接，单击此超链接会将相应的页码以 GET 方式传递到本页。

⑦ 第 30~36 行，用 SELECT…LIMIT 子句查询出本页中需要显示的记录，并用循环输出到一个表格中。

此程序虽然已经实现了分页功能，但是功能还不是很完善。比如本例中全部页码直接用循环列出，这样虽然很方便，但是在页码很多（如几百页）的情况下会出现版面混乱。可以使用“上一页”、“下一页”、“首页”、“尾页”等方法，简化页码显示。可以根据所学知识，尝试完善此程序的功能。

9.4　案例 9 编写一个 PHP+MySQL 留言本程序

9.4.1　案例分析

留言本是 Web 开发中比较典型的应用之一。留言本程序虽然简单，但是包括了表单提交、数据接收、写入数据库、读出数据库、分页等数据库编程的方方面面。能够熟练编写

一个功能较全面留言本是 Web 开发人员的基本功。本案例就来编写一个支持填写留言、浏览留言、管理员回复留言、删除留言等功能的留言本。

9.4.2 实施步骤

1. 创建数据库和表

本例中一共用到 1 个数据库和 2 个表。数据库名为 gbook。两个表为 Message 和 Admin。Message 表存放用户提交的留言以及管理员的回复。Admin 用来存放管理员的用户名和密码。两个表的字段设置和字段数据类型见表 9-2 和表 9-3。

表 9-2 Admin 表字段设置情况

字段名	类型	描述
username	varchar(20)	用户名
userpass	varchar(20)	用户密码

表 9-3 message 表字段设置情况

字段名	类型	描述
id	int(4)自增型	按顺序为每篇留言自动分配一个惟一的 ID 号
author	varchar(20)	留言人，20 字节以内
addtime	DATETIME	留言日期时间
content	varchar(1000)	留言内容。1000 字节以内
reply	varchar(1000)	管理员回复，1000 字节以内

现在数据库和表的设计已经完成，打开 phpMyAdmin，按照上述设计创建数据库和表。如果对命令提示符熟悉的话，可以尝试在 MySQL 控制台用命令的方式创建。这里给出 SQL 语句供读者参考。

创建数据库 gbook：

```
CREATE DATABASE 'gbook' ;
```

创建表 admin：

```
CREATE TABLE 'admin' (
'username' VARCHAR(20)NOT NULL ,
'userpass' VARCHAR(20)NOT NULL
)
```

创建表 message：

```
CREATE TABLE 'message' (
'id' INT(4)NOT NULL AUTO_INCREMENT PRIMARY KEY ,
'author' VARCHAR(20)NOT NULL ,
'addtime' DATETIME NOT NULL ,
```

```
'content' VARCHAR(1000)NOT NULL ,
'reply' VARCHAR(1000)NOT NULL
)
```

创建完成后，可以在 Admin 表中插入一条记录，也就是初始化一个管理员账号。不妨设置一个用户名为 Admin，密码为 adminpass 的管理员账号。如图 9-7 所示。



图 9-7　初始化管理员账号

2. 代码编写

本留言本一共用到 6 个页面。分别是：

- index.php：留言本首页；
- login.php：管理员登录页面；
- send.php：用户填写留言页面；
- manage.php：管理员管理页面；
- reply.php：管理员进行回复的页面；
- logout.php：注销登录的页面。

理论上来讲，这几个页面的编写顺序是无关紧要的。但是为了更好的说明本程序开发的流程，可以采用下面的步骤：

（1）编写 send.php，供用户提交留言，只有提交了留言才能进行后面的留言显示、留言管理等；

（2）编写 index.php，本页面将显示 10 条最近的留言，并有分页功能；

（3）编写 login.php，供管理员登录；

（4）编写 manage.php 和 reply.php，用来进行留言管理；

（5）编写 logout.php，用来注销登录。

由于程序本身是一个整体，先给出全部网页的完整代码，然后再运行，给出运行效果图。

注 意

> 由于版面有限，有些程序本来应该写在一行当中，却被迫分成两行。读者在读程序时应注意区分，另外应多多注意细节（引号、分号、逗号等）

程序清单　文档 send.php

```
<!—文件 send.php：用户提交留言页面-->
<?php
$name=$_POST["name"];
//看用户是否提交了新留言，如果提交了，则写入表 message
if($name!=""){
 $content=$_POST["content"];
 //以下代码获得当前日期/时间
 $addtime=date("Y-m-d h:i:s");
 $id=mysql_connect("localhost","root","1234");
 mysql_select_db("gbook",$id);
$query="insert into message(author,addtime,content,reply)values
('$name' ,'$addtime' ,'$content' ,'')";
 mysql_query($query,$id);
 mysql_close($id);
echo "<script language=javascript> alert('留言成功!单击确定查看留言.')
;location.href=' index.php';</script>";
 exit;
}
?>
<HTML>
<HEAD>
<TITLE>欢迎使用简约不简单-我的留言本</TITLE>
</HEAD>
<BODY>
<table border=1 cellspacing=0 cellspadding=0
style="border-collapse:collapse" align=center width=400
bordercolor=black>
<tr><td height=100 bgcolor=#6C6C6C>
<font style="font-size:30px" color=#ffffff face="黑体">简约不简单-我的原创留
言本</font>
</td>
<tr><td height=25>
 <a href=send.php>[我要写留言]</a>  <a href=login.php>[管理
留言]</a>
</td></tr>
<tr><td height=200>
  <form method="POST" action="send.php">
    <table border="1" width="95%" id="table1" cellspacing="0"
cellpadding="0" bordercolor="#808080"
style="border-collapse:collapse" height="265">
        <tr>
```

```
            <td colspan="2" height="29">
            <p align="center">欢迎您填写留言</td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">您的名字</td>
            <td width="67%">
<input type="text" name="name" size="20"></td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">留言内容</td>
            <td width="67%">
<textarea rows="10" name="content" cols="31"></textarea></td>
        </tr>
        <tr>
            <td width="99%" colspan="2">
            <p align="center"><input type="submit" value="提交"
        name="B1"></p>
            </td>
        </tr>
    </table>
    </form>
    </td></tr>
<tr><td height=80 bgcolor=#6c6c6c align=center>
<font color=#FFFFFF>版权所有：我的工作室<br>E-mail:rznqp@163.com
</td></tr>
</table>
</body>
</html>
```

本程序的大部分代码都是普通 HTML 代码，只有开头一部分是 PHP 代码。作用是将用户输入的表单数据接收并存入数据库。这样的例子在前面已经多次讲解，在此不再详述，请自行领会。

程序清单　文档 index.php

```
<!—文件 index.php：留言本首页-->
<HTML>
<HEAD>
<TITLE>欢迎使用简约不简单-我的留言本</TITLE>
<style type=text/css>
TD{
font-size:12px;
```

```
line-height:150%;
}
</style>
</HEAD>
<BODY>
<table border=1 cellspacing=0 cellspadding=0
style="border-collapse:collapse" align=center width=400 bordercolor=black
height="382">
<tr><td height=100 bgcolor=#6C6C6C
style="font-size:30px;line-height:30px" >
<font color=#ffffff face="黑体">简约不简单-我的留言本</font>
</td><tr>
<tr><td height=25>
 <a href=send.php>[我要写留言]</a>  <a href=login.php>
[管理留言]</a>
</td></tr>
<tr><td height=200>
  <?php
   //连接到数据库，查询留言并进行分页计算
   $id=mysql_connect("localhost","root","1234");
   mysql_select_db("gbook",$id);
   $query="select * from message ";
   $result=mysql_query($query,$id);
   if(mysql_num_rows($result)<1){
   echo " 目前数据表中还没有任何留言！";
   }else{ //ELSE 开始
     $totalnum=mysql_num_rows($result);
     $pagesize=10;
     $page=$_GET["page"];
     if($page==""){
       $page=1;
     }
     $begin=($page-1)* $pagesize;
     $totalpage=ceil($totalnum/$pagesize);
     //输出分页信息
     echo "<table border=0 width=95%><tr><td>";
     $datanum=mysql_num_rows($result);
     echo "共有留言".$totalnum."条。";
     echo "每页".$pagesize."条，共".$totalpage."页<br>";
     //输出页码
     for($j=1;$j<=$totalpage;$j++){
      echo "<a href=index.php?page=".$j.">[".$j."] </a>";
     }
```

```
      echo "<br>";
      //从 message 表中查询当前页所要显示的留言，并根据时间排序
  $query="SELECT * FROM message order by addtime desc limit
  $begin,$pagesize";
      $result=mysql_query($query,$id);
      //循环输出所有留言，如果管理员已经回复则同时输出回复
      for($i=1;$i<=$datanum;$i++){
       $info=mysql_fetch_array($result,MySQL_ASSOC);
       echo "->[".$info['author']."]于".$info['addtime']."说：<br>";
       echo "  ".$info['content']."<br>";
        if($info['reply']!=""){
        echo "<b>管理员回复：</b>".$info['reply']."<br>";
        }
       echo "<hr>";
       }
      echo "</table>";
    } //ELSE 结束
    mysql_close($id);
   ?>
   </td></tr>
   <tr><td height=60 bgcolor=#6c6c6c align=center>
   <font color=#FFFFFF>版权所有：我的工作室<br>E-mail:rznqp@163.com
   </td></tr>
   </table>
   </body>
   </html>
```

本程序虽然较长，但是大部分代码如分页、数据查询与排序等，在前面的章节中都有过介绍，而且关键代码都作了注释，请自行领会。

程序清单　文档 login.php

```
<!—文件 login.php：管理员登录页面-->
<?php
$name=$_POST["name"];
//如果用户提交了登录信息，则连接到数据库，与数据库中的信息相比较
if($name!=""){
 $password=$_POST["password"];
 $id=mysql_connect("localhost","root","1234");
 mysql_select_db("gbook",$id);
 $query="select * from admin where username='$name'";
 $result=mysql_query($query,$id);
 if(mysql_num_rows($result)<1){
 echo "该用户不存在!请重新登录!";
```

```
 }else{
 $info=mysql_fetch_array($result,MySQL_ASSOC);
 if($info['userpass']!=$password){
 echo "密码输入错误!请重新登录!";
 }else{
//如果用户名密码都正确，则注册一个 session 来标记其登录状态。
  session_start();
  $_SESSION["login"]="YES";
echo "<script language=javascript>alert('登录成功!');location.href=
'manage.php' ; </script>";
  exit;
   }
 }
 mysql_close($id);
}
?>
<HTML>
<HEAD>
<TITLE>欢迎使用简约不简单-我的留言本</TITLE>
</HEAD>
<BODY>
<table border=1 cellspacing=0 cellspadding=0
style="border-collapse:collapse" align=center width=400 bordercolor=
black height="358">
<tr><td height=100 bgcolor=#6C6C6C>
<font style="font-size:30px" color=#ffffff face="黑体">简约不简单-我的留言本
</font>
</td>
<tr><td height=25>
 <a href=send.php>[我要写留言]</a>  <a href=login.php>
[管理留言]</a>
</td></tr>
<tr><td height=178>
  <form method="POST" action="login.php">
<table border="1" width="95%" id="table1" cellspacing="0" cellpadding="0"
bordercolor="#808080" style="border-collapse:collapse" height="154">
        <tr>
            <td colspan="2" height="29">
            <p align="center">欢迎管理员登录</td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">用户名</td>
```

```
            <td width="67%"><input type="text" name="name" size="20"></td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">密  码</td>
        <td width="67%"><input type="password" name="password"
    size="20"></td>
        </tr>
        <tr>
            <td width="99%" colspan="2">
        <p align="center"><input type="submit" value="登录" name="B1"></p>
            </td>
        </tr>
    </table>
    </form>
    </td></tr>
<tr><td height=58 bgcolor=#6c6c6c align=center>
<font color=#FFFFFF>版权所有：我的工作室<br>E-mail:rznqp@163.com
</td></tr>
</table>
</body>
</html>
```

本页中也是 HTML 代码占了绝大部分篇幅。核心 PHP 代码只有前面部分。此程序完成了登录验证，可以参考此程序来领会用 session 实现用户登录的方法。

程序清单　文档 manage.php

```
<!—文件 manage.php：管理员对留言进行管理页面-->
<?php
//首先判断用户是否已经登录。因为只有登录用户才能访问此页面。
session_start();
if($_SESSION["login"]!="YES"){
 echo "你还没有登录,无法管理留言!";
 exit;
}
//如果用户提交了删除留言的 ID，则删除此留言
$delid=$_GET["delid"];
if($delid!=""){
 $id=mysql_connect("localhost","root","1234");
 mysql_select_db("gbook",$id);
 mysql_query("delete from message where id=".$delid);
 echo "<script language=javascript>alert('删除成功!');</script>";
 mysql_close($id);
```

```
}
?>
<HTML>
<HEAD>
<TITLE>欢迎使用简约不简单-我的留言本</TITLE>
<style type=text/css>
TD{
font-size:12px;
line-height:150%;
}
</style>
</HEAD>
<BODY>
<table border=1 cellspacing=0 cellspadding=0
style="border-collapse:collapse" align=center width=400 bordercolor=
black height="382">
<tr><td height=100 bgcolor=#6C6C6C
style="font-size:30px;line-height:30px" >
<font color=#ffffff face="黑体">简约不简单-我的留言本</font>
</td>
<tr><td height=25>
  <a href=logout.php>[注销登录]</a>
</td></tr>
<tr><td height=200>
  <?php
//数据查询、分页、输出部分与 index.php 基本相同
   $id=mysql_connect("localhost","root","1234");
   mysql_select_db("gbook",$id);
   $query="select * from message";
   $result=mysql_query($query,$id);
   if(mysql_num_rows($result)<1){
   echo " 目前数据表中还没有任何留言！";
   }else{ //ELSE 开始
     $totalnum=mysql_num_rows($result);
     $pagesize=10;
     $page=$_GET["page"];
     if($page==""){
       $page=1;
     }
     $begin=($page-1)* $pagesize;
     $totalpage=ceil($totalnum/$pagesize);
     echo "<table border=0 width=95%><tr><td>";
     $datanum=mysql_num_rows($result);
```

```
        echo "共有留言".$totalnum."条。";
        echo "每页".$pagesize."条，共".$totalpage."页<br>";
        for($j=1;$j<=$totalpage;$j++){
         echo "<a href=manage.php?page=".$j.">[".$j."] </a>";
        }
        echo "<br>";

  $query="SELECT * FROM message  order by addtime desc limit
  $begin,$pagesize";
        $result=mysql_query($query,$id);
        for($i=1;$i<=$datanum;$i++){
         $info=mysql_fetch_array($result,MySQL_ASSOC);
         echo "->[".$info['author']."]于".$info['addtime']."说：<br>";
         echo "  ".$info['content']."<br>";
          if($info['reply']!=""){
          echo "<b>管理员回复：</b>".$info['reply']."<br>";
          }
          //此处增加了删除和回复功能
         echo "[<a href=manage.php?delid=".$info['id'].">删除此留言</a>]
       [<a href=reply.php?id=".$info['id'].">回复留言</a>]";
         echo "<hr>";
         }
        echo "</table>";
     } //ELSE 结束
     mysql_close($id);
    ?>
   </td></tr>
   <tr><td height=60 bgcolor=#6c6c6c align=center>
   <font color=#FFFFFF>版权所有：我的工作室<br>E-mail:rznqp@163.com
   </td></tr>
   </table>
   </body>
   </html>
```

本程序与 index.php 即留言本首页的很多代码都很接近，只是增加了删除代码和回复按钮。单击“回复”按钮，将打开回复留言的页面，并将要回复的留言的编号传递过去。

程序清单　文档 reply.php

```
<!--文件 reply.php：对留言进行回复的页面-->
<?php
session_start();
if($_SESSION["login"]!="YES"){
 echo "你还没有登录,无法管理留言!";
```

```
 exit;
}
$msgid=$_GET["id"];
$reply=$_POST["reply"];
//如果提交了回复，则打开数据库，将留言的 reply 字段值设置为回复内容
if($reply!=""){
 $id=mysql_connect("localhost","root","1234");
 mysql_select_db("gbook",$id);
 $query="update message set reply='$reply' where id=".$msgid;
 mysql_query($query,$id);
 echo "<script language=javascript>alert('回复成功!'); location.href
='manage.php'; </script>";
 exit;
 mysql_close($id);
}

 $id=mysql_connect("localhost","root","1234");
 mysql_select_db("gbook",$id);
 $query="select * from  message where id=$msgid";
 $result=mysql_query($query,$id);
 if(mysql_num_rows($result)<1){
 echo "没有此留言";
 exit;
 }
 $msg=mysql_fetch_array($result,MySQL_ASSOC);
?>
<HTML>
<HEAD>
<TITLE>欢迎使用简约不简单-我的留言本</TITLE>
</HEAD>
<BODY>
<table border=1 cellspacing=0 cellspadding=0
style="border-collapse:collapse" align=center width=400 bordercolor=black
height="358">
<tr><td height=100 bgcolor=#6C6C6C>
<font style="font-size:30px" color=#ffffff face="黑体">简约不简单-我的留言本
</font>
</td>
<tr><td height=25>
  <a href="manage.php">[返回管理]</a>
</td></tr>
<tr><td height=178>
  <form method="POST" action="reply.php?id=<?php echo $msgid;?>">
```

```
    <table border="1" width="95%" id="table1" cellspacing="0"
cellpadding="0" bordercolor="#808080"
style="border-collapse:collapse"
height="154">
        <tr>
            <td colspan="2" height="29">
            <p align="center">管理员回复留言</td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">留言 ID</td>
            <td width="67%"><?php echo $msg['id'];?></td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">留言人</td>
            <td width="67%"><?php echo $msg['author'];?></td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">留言时间</td>
            <td width="67%"><?php echo $msg['addtime'];?></td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">留言内容</td>
            <td width="67%"><?php echo $msg['content'];?></td>
        </tr>
        <tr>
            <td width="32%">
            <p align="right">请输入回复</td>
        <td width="67%"><textarea rows="7" name="reply"
    cols="33"></textarea></td>
        </tr>
        <tr>
            <td width="99%" colspan="2">
        <p align="center"><input type="submit" value="确定" name="B1"></p>
        </td>
        </tr>
    </table>
    </form>
    </td></tr>
<tr><td height=58 bgcolor=#6c6c6c align=center>
```

```
<font color=#FFFFFF>版权所有：我的工作室<br>E-mail:rznqp@163.com
</td></tr>
</table>
</body>
</html>
<?php mysql_close(); ?>
```

本程序代码比较简单，不再详细讲解。

最后一个页面是注销登录的页面，这个代码最为简单。只需要将 Session 值设置为空，则登录状态撤销。

程序清单 文档 logout.php

```
<!—文件 logout.php：注销登录页面-->
<?php
session_start();
$_SESSION["login"]="";
echo "已成功退出。[<a href=index.php>回首页</a>]";
exit;
?>
```

到此为止，已经给出了这个留言本程序的全部代码。虽然代码总篇幅比较长，但是普通 HTML 代码占了大部分篇幅，真正 PHP 核心代码并不多。只要细心研读，都能熟练掌握。下面是留言本的运行结果。

首先打开 index.php。此时由于还没有任何留言，所以页面上提示“目前数据表中还没有任何留言”，如图 9-8 所示。

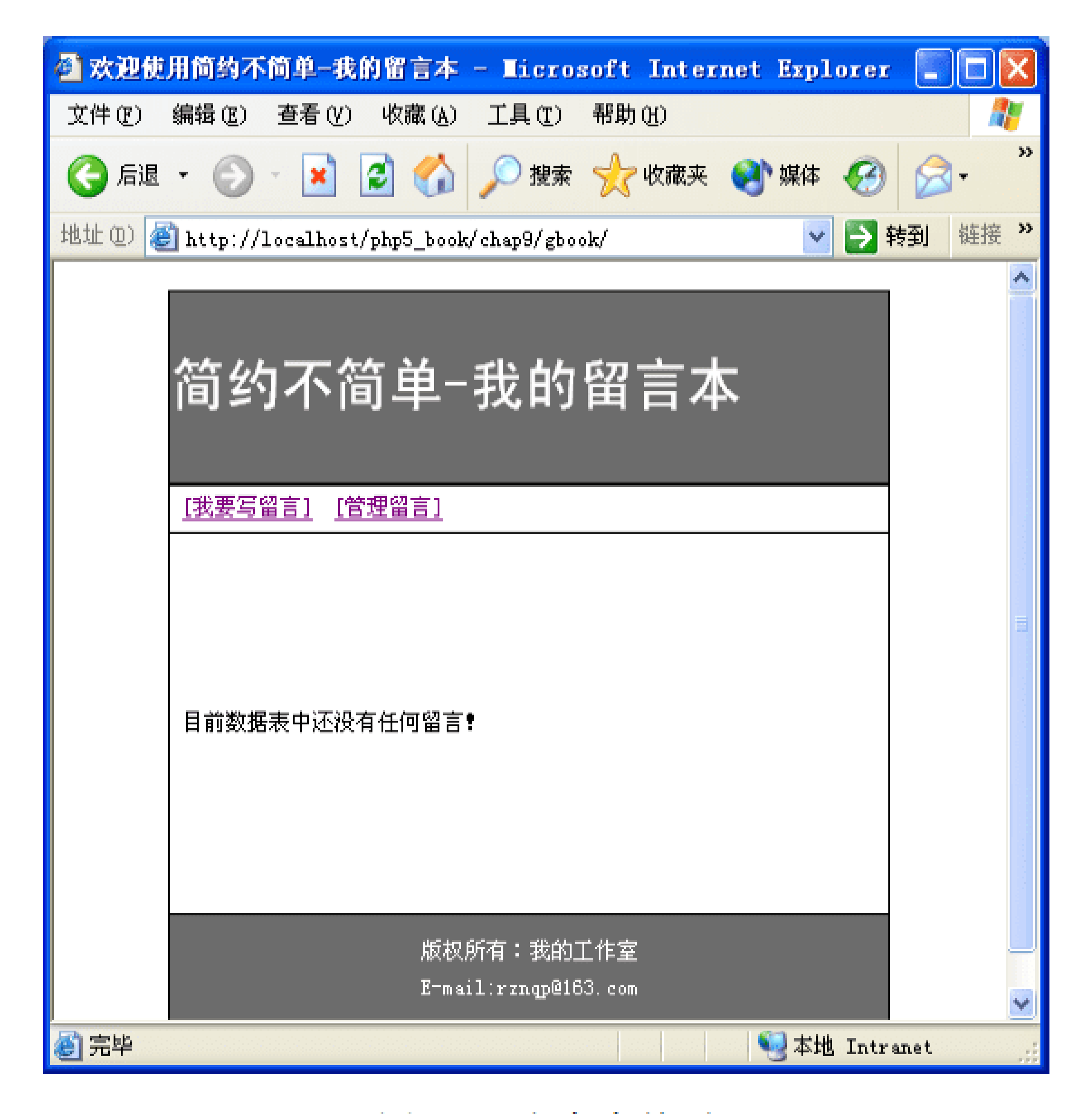


图 9-8 留言本首页

单击“我要写留言”，打开 send.php，向里面添加一条留言，如图 9-9 所示。

单击“提交”，会提示“留言提交成功”并自动转到首页。为了更好的测试出效果，反复运行 send.php，多提交几条留言。然后再回到 index.php，可以看到多条留言按照发布时间的顺序从高到低排列。如图 9-10 所示。

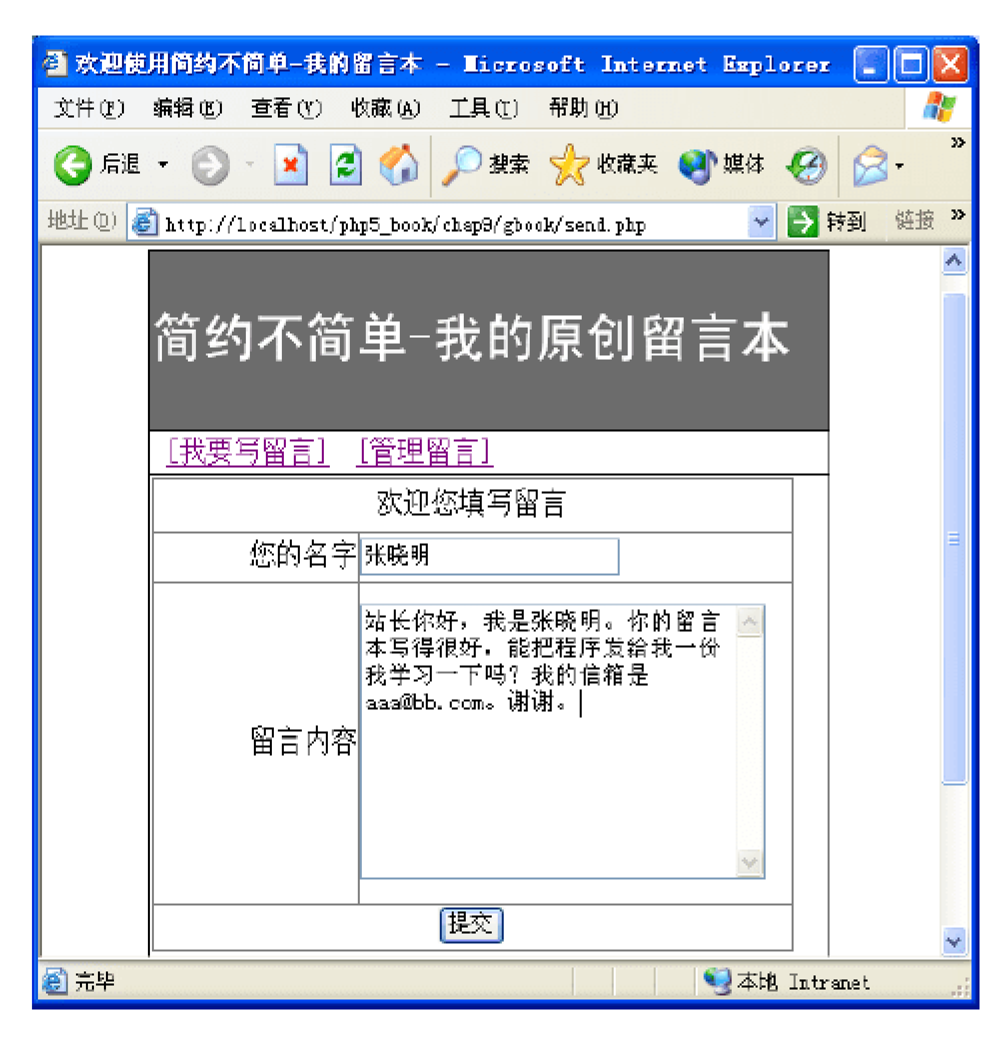


图 9-9　提交留言

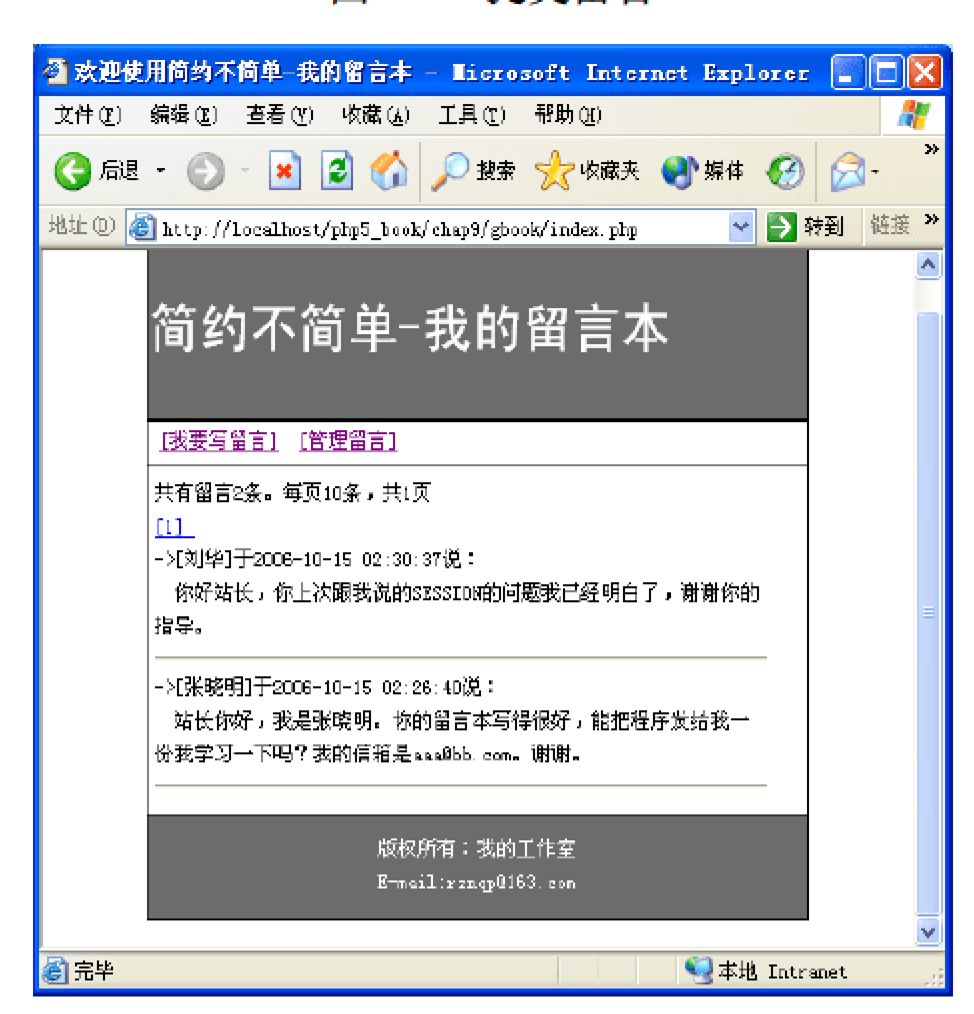


图 9-10　留言本首页

单击“留言管理”，打开管理员登录页面 login.php。如图 9-11 所示。在此页面中输入事先设置好的账号，用户名是 admin，密码是 adminpass。如果输入有误，则会给出提示。输入正确则自动转到 manage.php 进行留言管理。如图 9-12 所示。

单击“删除留言”，将会删除该留言。单击“回复留言”，将打开 reply.php 留言回复页面。如图 9-13 所示。输入回复信息后单击“确定”，将提示回复成功，并自动转回 manage.php。

这时已经可以看到回复的结果，如图 9-14 所示。此时在 index.php 同样也可以看到回复结果。

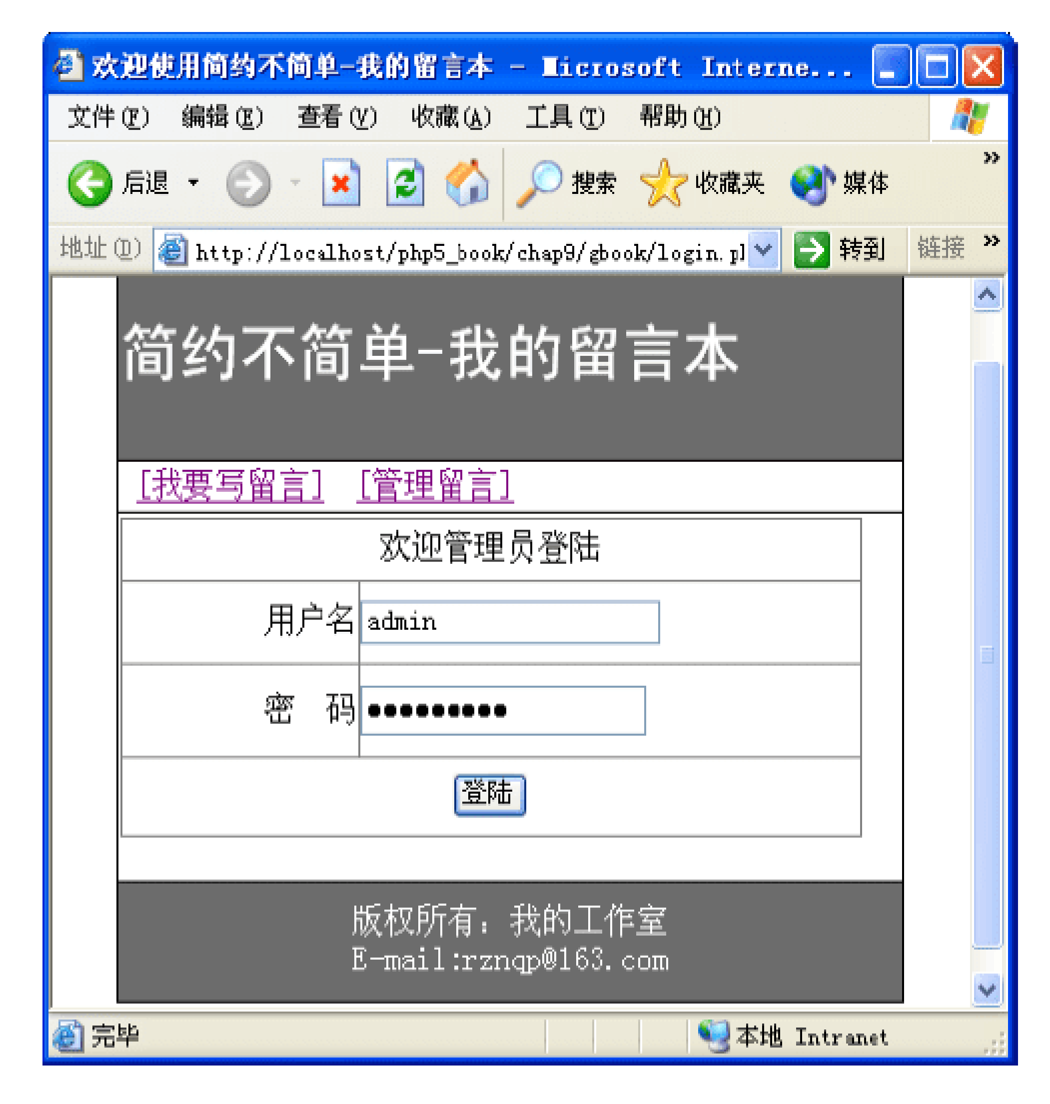


图 9-11 管理员登录界面

图 9-12 管理员管理界面

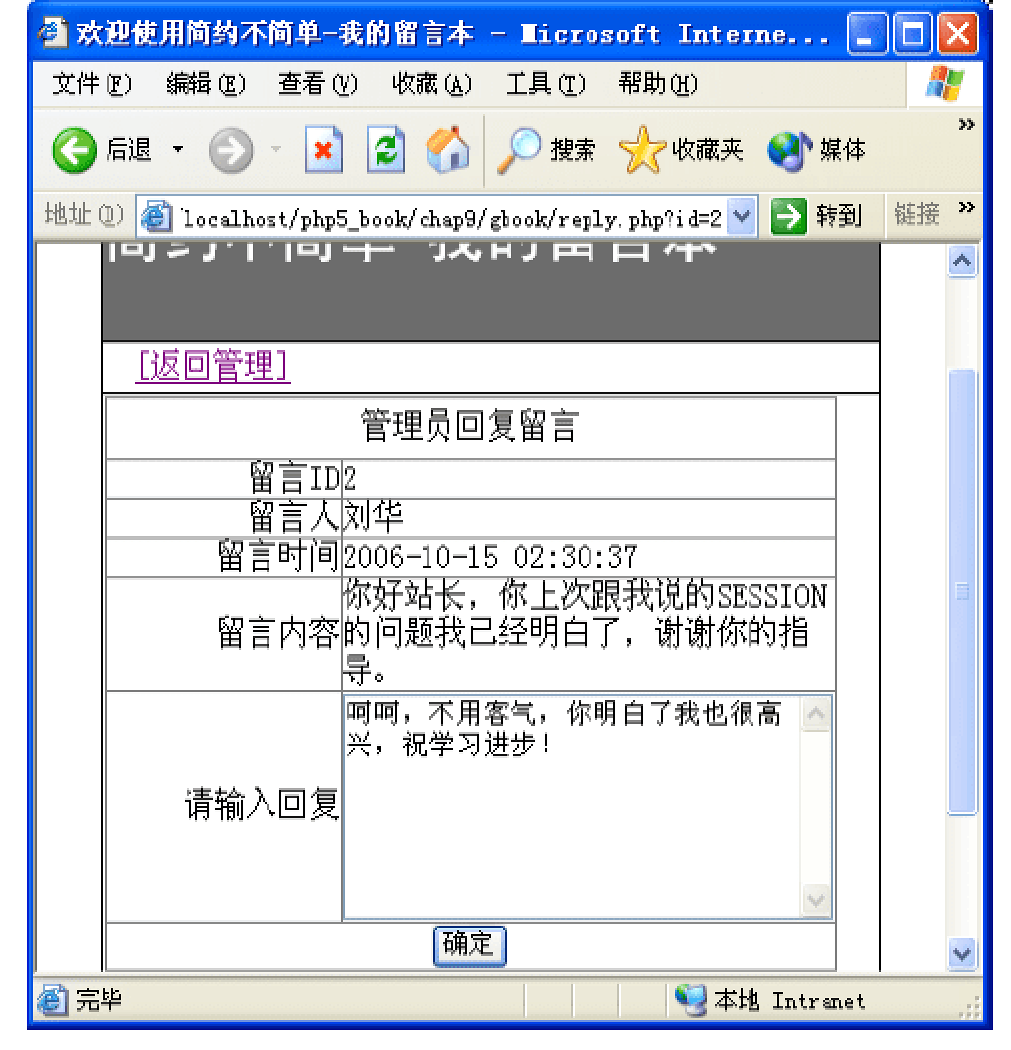


图 9-13 回复留言

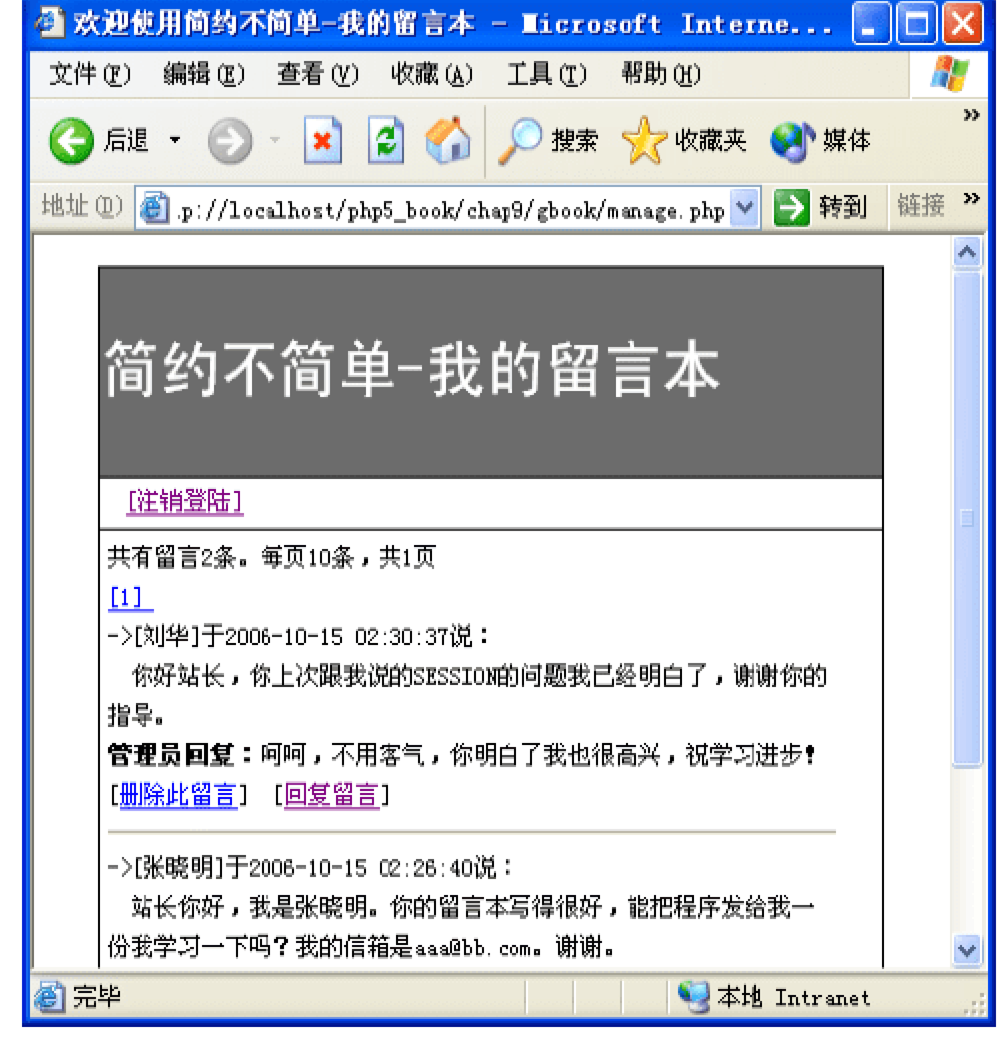


图 9-14 显示留言和回复

最后，可以通过单击“注销登录”来解除登录状态。单击后提示“已成功退出”。

至此，留言本开发完成。实际上，这个留言本还有值得改进的地方，比如管理员可以增加修改密码功能，普通用户可以注册以后才能发表留言等。可以利用所学知识，动手完善此程序。

9.5 本 章 小 结

本章中主要介绍了PHP+MySQL编程的具体方法和步骤。首先介绍了PHP连接MySQL的原理和流程，然后介绍了连接前的注意事项，最后详细讲解了PHP操作MySQL数据库

的方法。在实例中给出了一个留言本的全部代码，来说明 PHP+MySQL 数据库编程的思路和方法。通过本章的学习，应熟练掌握 PHP 操作 MySQL 的方法，熟练进行输入的插入、删除、修改和查询操作。

9.6　练 习 题

1. PHP 连接 MySQL 前需要做哪些准备工作？
2. 常用的 PHP 操作 MySQL 的函数有哪些？
3. 在 PHP 中如何对多条记录进行分页？
4. 如果要为案例 9 的留言本程序增加显示留言者 IP 地址的功能，数据库和程序需要做哪些改动？并请上机实现之。

CHAPTER 10

PHP 5 程序开发综合范例

本章重点

- PHP+MySQL 项目开发流程
- 综合范例中的数据库设计
- PHP+MySQL 开发的常用技术

PHP + MySQL

PHP+MySQL 的组合，可以开发几乎所有流行的 Web 应用系统。如论坛、留言本、新闻与文章管理系统、各类信息发布系统、投票程序等等。其中论坛系统具有很强的代表性。一个论坛系统几乎包含了 PHP+MySQL 编程的方方面面，如果能够掌握 PHP+MySQL 论坛开发技术，那么开发其他应用系统的也必然不成问题。任何浏览者都有浏览任何模块、任何帖子以及任何回复的权限，而且任何浏览者都有权利在本论坛注册。

下面将用整章的篇幅来介绍一个简易 BBS 系统的开发过程。虽然这是一个简易的 BBS，但是它基本囊括了一个论坛系统所应具备的所有基本功能，是一个有一定参考价值的系统。以下将给出这个论坛系统的全部核心代码。仅将给出的代码作为一个参考，自己手工来编写相关代码。这样可以锻炼自己复杂代码的编写能力，为将来完全独立地开发 PHPWeb 应用系统打下基础。

10.1　需 求 分 析

任何浏览者都有浏览任何模块、任何帖子以及任何回复的权限，而且任何浏览者都有权利注册成为本论坛的用户。

任何注册用户都有浏览任何模块、任何帖子以及任何回复的权限，而且还有发新帖子、回复任何帖子的权限。任何注册用户都有被设为某子模块版主的机会。

版主有浏览任何模块、任何帖子以及任何回复的权限，而且还有发新帖子、回复任何帖子的权限。版主还有删除任模块的帖子和回复的删除权限。

管理员除了有版主所有的权限外。还有添加父模块、管理父模块、添加子模块、管理子模块、查看所有用户、删除用户、检索用户、自己密码更改等权限。

10.2　流程设计

如图 10-1 所示是简单的流程设计及功能权限分配。

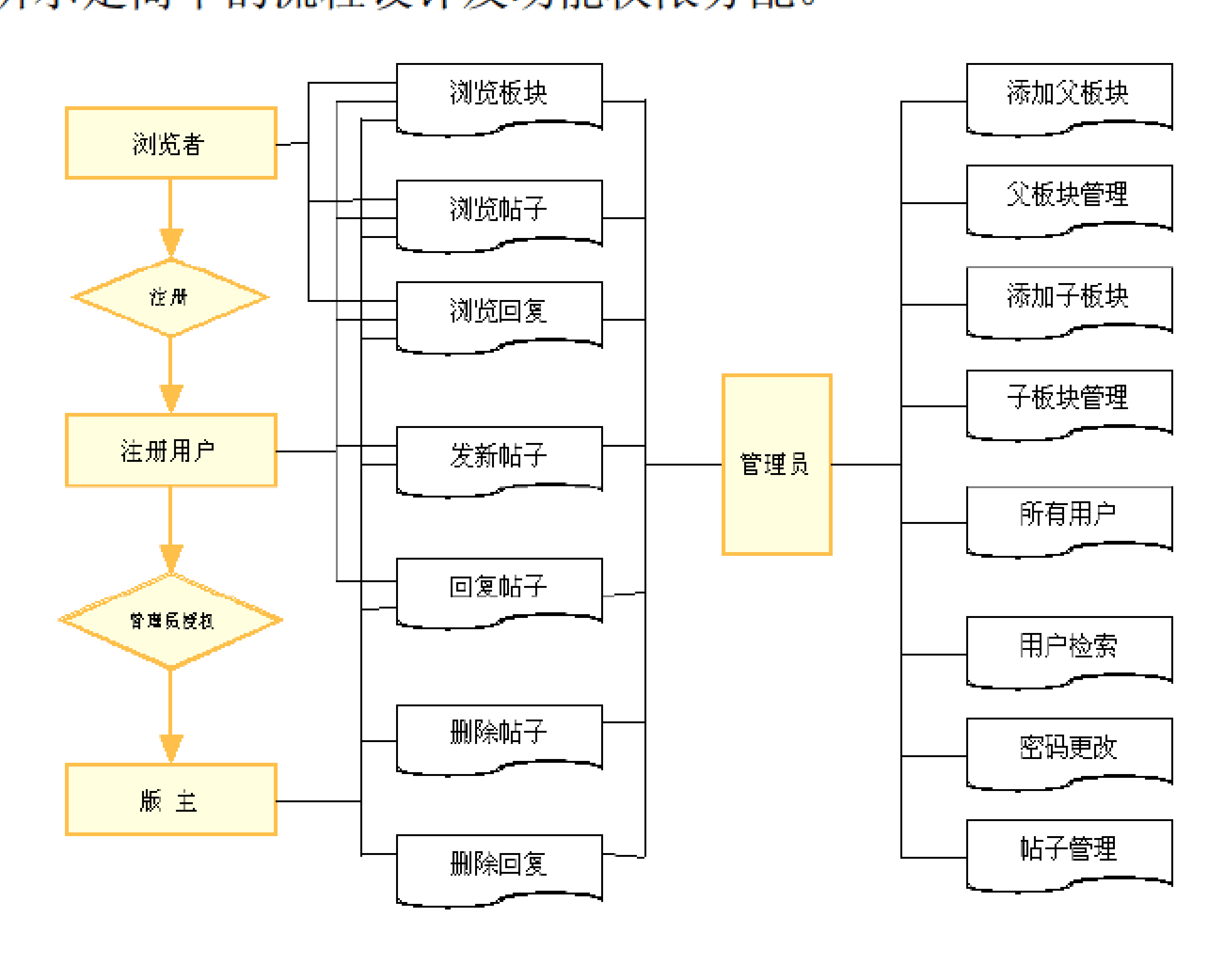


图 10-1　论坛系统流程设计

10.3 数据库设计

本论坛的数据库名为“bbs_data”，共含 5 个数据表，这 5 个数据表之间的关系简单描述如图 10-2 所示（仅画出有关联关系的字段）。

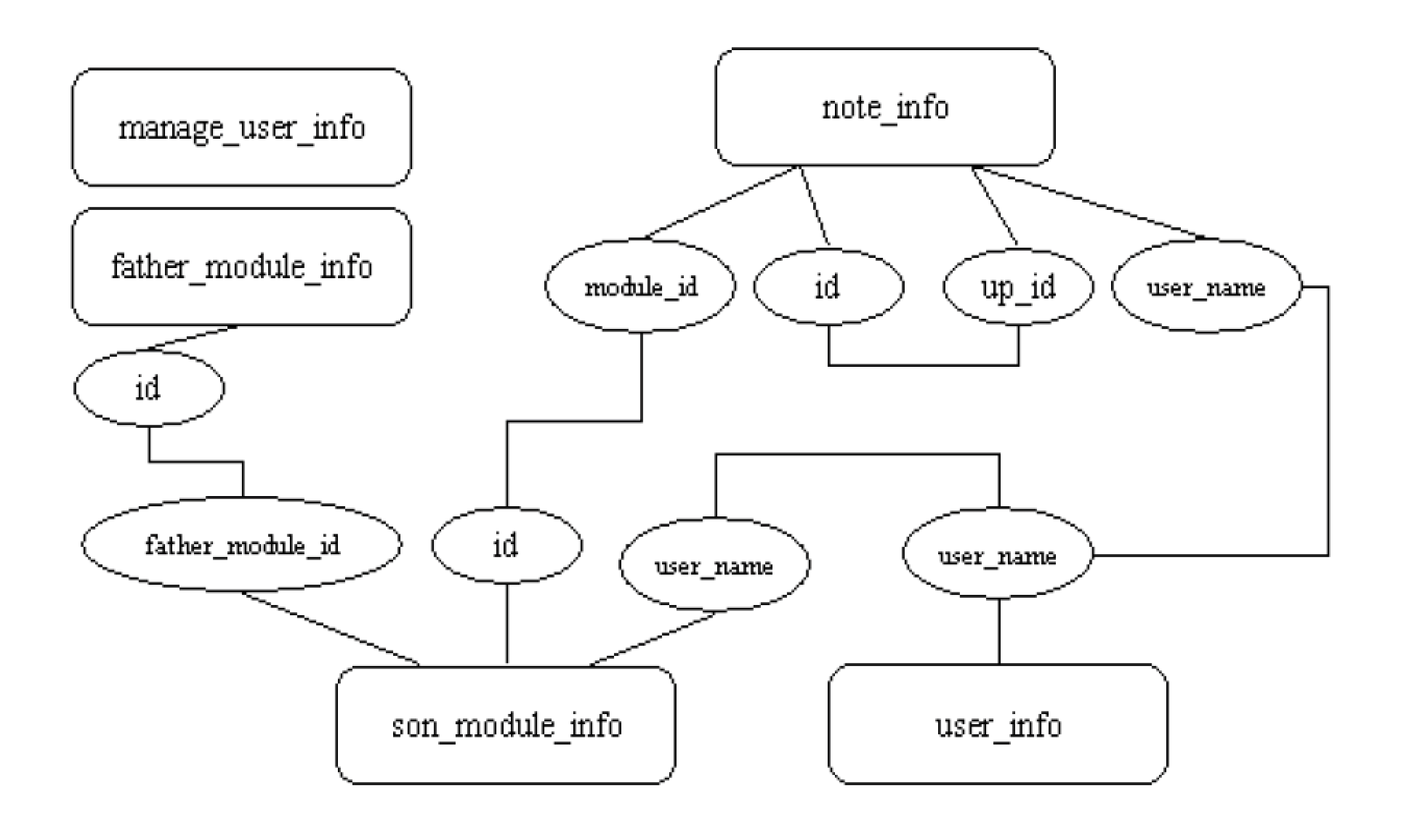


图 10-2 数据库设计

其详细的字段设计如表 10-1~表 10-5 所示。

表 10-1 manage_user_info（管理用户信息数据表）

编号	字段名	类型	字段意义	备注
1	id	int		
2	user_name	char(16)	管理用户登录名	
3	user_pw	char(16)		

表 10-2 user_info（用户信息数据表）

编号	字段名	类型	字段意义	备注
1	id	int		
2	user_name	char(16)	管理用户登录名	
3	user_pw	char(16)		
4	time1	datetime	注册时间	
5	time2	datetime	最后登录时间	

表 10-3　father_module_info（父模块信息数据表）

编号	字段名	类型	字段意义	备注
1	id	int		
2	module_name	char(66)	模块名称	
3	show_order	int	显示序号	

表 10-4　son_module_info（子模块信息数据表）

编号	字段名	类型	字段意义	备注
1	id	int		
2	father_module_id	int	隶属的大模块的 id	同 father_module_info 中 id
3	module_name	char(66)	子模块名称	
4	module_cont	Text	子模块简介	
5	user_name	char(16)	本版版主	同 user_info 中的 user_name

表 10-5　note_info（发帖信息数据表）

编号	字段名	类型	字段意义	备注
1	id	Int		
2	module_id	int	隶属的子模块的 id	同 son_module_info 中 id
3	up_id	int	回复帖子的 id	同本表中的 id
4	title	char(88)	帖子标题	
5	cont	text	帖子内容	
6	time	datetime	发帖时间	
7	user_name	char(16)	发帖用户名	同 user_info 中的 user_name
8	times	int	浏览次数	

10.4　代码编写

10.4.1　论坛的程序清单与存放路径

本简易论坛的所有程序清单及存放路径如表 10-6 所示。

表 10-6　论坛程序清单

根目录下	子目录	备注
inc		文件夹：存放被包含文件
	foot.php	版权文件
	head.php	头文件
	myfunction.inc	自编函数库文件
	mysql.inc	自编连接数据库、服务器、sql 语句执行函数库
	style.css	论坛 CSS 样式表
	total_info.php	论坛总信息
manage		文件夹：存放后台管理的所有文件
	image	文件夹：存放后台管理界面相关的图片
	Admin_Style.css	后台管理中页面所用的 CSS
	bottom.php	后台管理的版权文件
	father_module_add.php	父模块添加页
	father_module_bj.php	父模块编辑页
	father_module_list.php	父模块列表页
	fun_head.php	后台管理中右侧页面中的头文件（用函数构造）
	index.php	后台管理的主页面
	index_left.php	后台管理主页面所包含的左侧的页面
	index_right.php	后台管理主页面所包含的右侧的下部页面
	index_top.php	后台管理主页面所包含的右侧的上部页面
	left.css	后台管理主页面中左侧页面的 CSS
	login.php	管理用户登录页面
	menu.js	实现菜单的 JavaScript 文件
	session.inc	校验 Session 是否存在的页面
	son_module_add.php	子模块添加页面
	son_module_bj.php	子模块编辑页面
	son_module_list.php	子模块列表页面
	user_js.php	用户检索页面
	user_list.php	显示所有用户列表的页面
	user_pw_change.php	管理员密码更改页面
pic_sys		文件夹：存放本系统界面相关的图片（后台管理界面的图片除外）
系统文档		文件夹：存放系统设计或说明的有关文档
	数据库设计.doc	存放本系统的详细的数据库设计
	关系图.sdr	存放本系统数据库中数据表之间的关联关系的关系图

（续表）

根目录下	子目录	备注
index.php	用户访问的首页	
install.php	系统的安装文件	
module_list.php	某一子模块的帖子列表页面	
new_note.php	发布新帖的页面	
note_show.php	显示帖子和相关回复的页面	
register.php	新用户的注册页面	

10.4.2　论坛的主要程序代码

主要文件的主要代码如下（按编写的先后顺序）。

程序清单　文档 mysql.inc

```
<?php
class mysql{
    //连接服务器、数据库以及执行 SQL 语句的类库
    public $database;
    public $server_username;
    public $server_userpassword;
    function mysql()
     {  //构造函数初始化所要连接的数据库
       $this->server_username="root";
       $this->server_userpassword="root";
       }//end mysql()
     function link($database)
     {  //连接服务器和数据库
       //设置所有连接的数据库
        if($database==""){
            $this->database="bbs_data";
            }else{
            $this->database=$database;
            }
        //连接服务器和数据库
         if(@$id=mysql_connect('localhost',$this->server_username,$this->>
    server_userpassword)){
          if(!mysql_select_db($this->database,$id)){
                echo "数据库连接错误!!! ";
               exit;
               }
           }else{
               echo "服务器正在维护中，请稍后重试!!! ";
```

```
                exit;
            }
    }//end link($database)
    function excu($query)
    {     //执行 SQL 语句
        if($result=mysql_query($query)){
                return $result;
        }else{
            echo "sql 语句执行错误!!! 请重试!!!";
                exit;
        }
    } //end  exec($query)
} //end class mysql
?>
```

程序清单　文档 myfunction.inc

```
<?php
class myfunction{
////////////////字符转换：向数据库中插入或更新时用////////////////////////
      function str_to($str)
      {
        $str=str_replace(" "," ",$str);  //把空格替换 html 的字符串空格
        $str=str_replace("<","&lt;",$str);  //把 html 的输出标记正常输出
        $str=str_replace(">","&gt;",$str);  //把 html 的输出标记正常输出
        $str=nl2br($str);                //把回车替换成 html 中的 br
        return $str;
          }
////////////////////由子模块的 id 返回该子模块的主题数//////////////////////
    function son_module_idtonote_num($son_module_id){
        $aa=new mysql;
        $aa->link("");
        $query="select * from note_info where module_id='$son_module_id'
    and up_id='0'";
        $rst=$aa->excu($query);
        return mysql_num_rows($rst);
    }
////////////////////由子模块的 id 返回该子模块的帖子数//////////////////////
    function son_module_idtonote_num2($son_module_id){
     //由子模块的 id 返回该子模块的主题数
        $aa=new mysql;
        $aa->link("");
$query="select * from note_info where module_id='$son_module_id'
and up_id='0'";
```

```
        $rst=$aa->excu($query);
        $num=mysql_num_rows($rst);
        while($note=mysql_fetch_array($rst,MySQL_ASSOC)){
      $query="select * from note_info where up_id='$note[id]'
  and module_id='0'";
            $rst=$aa->excu($query);
            $num+=mysql_num_rows($rst);
            }
            return $num;
    }
////////////////////由子模块的 id 输出该子模块的最新帖子////////////////////
    function son_module_idtolast_note($son_module_id){
     //由子模块的 id 输出该子模块的最新帖子
        $aa=new mysql;
        $aa->link("");
$query="select * from note_info where module_id='$son_module_id' order
by time desc limit 0,1";
        $rst=$aa->excu($query);
        $note=mysql_fetch_array($rst,MySQL_ASSOC);
            $query2="select * from note_info where id='$note[up_id]'";
            $rst2=$aa->excu($query);
            $note2=mysql_fetch_array($rst2,MySQL_ASSOC);
            echo $note2[title];
            echo "<br>";
            echo $note[time]."  ".$note[user_name];
    }
////////////////////由子模块的 id 输出该子模块的版主/////////////////////
    function son_module_idtouser_name($son_module_id){
     //由子模块的 id 输出该子模块的版主
        $aa=new mysql;
        $aa->link("");
        $query="select * from son_module_info where id='$son_module_id'";
        $rst=$aa->excu($query);
        $module=mysql_fetch_array($rst,MySQL_ASSOC);
        if($module[user_name]==""){
            return "版主暂缺";
            }else{
                return $module[user_name];
                }
    }
///////////////输出所有模块的下拉列表（子模块有参数）///////////////////
    function son_module_list($son_module_id){
     //输出所有模块的下拉列表（子模块有参数）
```

```
        $aa=new mysql;
        $aa->link("");
        $query="select * from father_module_info order by id";
        $rst=$aa->excu($query);
        echo "<select name=module_id>";
        while($father_module=mysql_fetch_array($rst,MySQL_ASSOC)){
            echo "<option value=>".$father_
        module[module_name]."</option>";
$query="select * from son_module_info where
father_module_id='$father_module[id]' order by id ";
            $rst2=$aa->excu($query);
            while($son_module=mysql_fetch_array($rst2,MySQL_ASSOC)){
echo "<option value=".$son_module[id].">     " .
$son_module[module_name] . "</option>";
                }
            }
        echo "</select>";
    }
////////////////////输出父模块的下拉列表//////////////////////////////
    function father_module_list($father_module_id){
    //输出父模块的下拉列表
        $aa=new mysql;
        $aa->link("");
        echo "<select name=father_module_id>";
        if($father_module_id==""){
            echo "<option selected>请选择...</option>";
            }else{
$query="select * from father_module_info where id='$father_module_id'";
                $rst=$aa->excu($query);
                $father_module=mysql_fetch_array($rst,MySQL_ASSOC);
echo "<option value=".$father_module[id].">".$father_module
[module_name]."</option>";
                }
        $query="select * from father_module_info order by show_order";
        $rst=$aa->excu($query);
        while($father_module=mysql_fetch_array($rst,MySQL_ASSOC)){
echo "<option value=".$father_module[id].">".$father_module[module_name].
"</option>";
            }
        echo "</select>";
    }
////////////////////由帖子的id返回该帖子被浏览的次数/////////////////////
    function note_idtotimes($note_id){
```

```
        $aa=new mysql;
        $aa->link("");
        $query="select * from note_info where id='$note_id'";
        $rst=$aa->excu($query);
        $note=mysql_fetch_array($rst,MySQL_ASSOC);
        return $note[times];
    }
/////////////////////由帖子的 id 返回该帖子的标题//////////////////////
    function note_idtotitle($note_id){
        $aa=new mysql;
        $aa->link("");
        $query="select * from note_info where id='$note_id'";
        $rst=$aa->excu($query);
        $note=mysql_fetch_array($rst,MySQL_ASSOC);
        return $note[title];
    }
/////////////////////由帖子的 id 返回帖子的回复数/////////////////////
    function note_idtonote_num($note_id){
        $aa=new mysql;
        $aa->link("");
        $query="select * from note_info where up_id='$note_id'";
        $rst=$aa->excu($query);
        $num=mysql_num_rows($rst);
        return $num+1;
    }
///////////////////由帖子的 id 输出帖子的最后回复时间///////////////////
    function note_idtolast_time($note_id){
        $aa=new mysql;
        $aa->link("");
        $query="select * from note_info where up_id='$note_id' order by
    time desc limit 0,1";
        $rst=$aa->excu($query);
        $note=mysql_fetch_array($rst,MySQL_ASSOC);
        echo $note[time];
    }
/////////////////////由帖子的 id 输出帖子的最后回复人/////////////////////
    function note_idtolast_user_name($note_id){
        $aa=new mysql;
        $aa->link("");
$query="select * from note_info where up_id='$note_id' order
by time desc limit 0,1";
        $rst=$aa->excu($query);
        $note=mysql_fetch_array($rst,MySQL_ASSOC);
```

```
        echo $note[user_name];
    }
////////////////////由子模块的 id 返回其父模块的名称//////////////////////
    function son_module_idtofather_name($son_module_id){
        $aa=new mysql;
        $aa->link("");
        $query="select * from son_module_info where id='$son_module_id'";
        $rst=$aa->excu($query);
        $module=mysql_fetch_array($rst,MySQL_ASSOC);
$query2="select * from father_module_info where
id='$module[father_module_id]'";
        $rst2=$aa->excu($query2);
        $module2=mysql_fetch_array($rst2,MySQL_ASSOC);
        return $module2[module_name];
    }
////////////////////由子模块的 id 返回本模块的名称//////////////////////
    function son_module_idtomodule_name($son_module_id){
        $aa=new mysql;
        $aa->link("");
        $query="select * from son_module_info where id='$son_module_id'";
        $rst=$aa->excu($query);
        $module=mysql_fetch_array($rst,MySQL_ASSOC);
        return $module[module_name];
    }
////////////////////所有帖子的总数//////////////////////
    function note_total_num(){
        $aa=new mysql;
        $aa->link("");
        $query="select * from note_info";
        $rst=$aa->excu($query);
        return mysql_num_rows($rst);
    }
////////////////////所有会员的总数//////////////////////
    function user_total_num(){
        $aa=new mysql;
        $aa->link("");
        $query="select * from user_info";
        $rst=$aa->excu($query);
        return mysql_num_rows($rst);
    }
////////////////////所有会员的总数//////////////////////
    function last_username(){
        $aa=new mysql;
```

```
        $aa->link("");
        $query="select * from user_info order by id desc limit 0,1";
        $rst=$aa->excu($query);
        $user=mysql_fetch_array($rst,MySQL_ASSOC);
        return $user[user_name];
    }
/////////////分页函数///////////////////
     function page($query,$page_id,$add,$num_per_page){
    //  include "mysql.inc";
    ////////使用方法为:
    ///////  $myf=new myfunction;
    ///////  $query="";
     ///////  $myf->page($query,$page_id,$add,$num_per_page);
    ///////  $bb=$aa->excu($query);
      $bb=new mysql;
      global $query;      //声明全局变量
      $bb->link("");
      $page_id=$_GET[page_id];    //接受 page_id
      if($page_id==""){
          $page_id=1;
          }
      $rst=$bb->excu($query);
      $num=mysql_num_rows($rst);
      if($num==0){
          echo "没有查到相关记录或没有相关回复! <br>";
          }
      $page_num=ceil($num/$num_per_page);
      for($i=1;$i<=$page_num;$i++){
          echo " [<a href=?".$add."page_id=$i>".$i."</a>]";
          }
      $page_up=$page_id-1;
      $page_down=$page_id+1;

      if($page_id==1){
echo "<a href=?".$add."page_id=$page_down>下一页</a>  
第".$page_id."页,共".$page_num."页";
          }
      else if($page_id>=$page_num-1){
           echo "<a href=?".$add."page_id=$page_up>上一页</a>  
      第".$page_id."页,共".$page_num."页";
           }
      else{
           echo "<a href=?".$add."page_id=$page_up>上一页</a>  
```

```
        <a href=?".$add."page_id=$page_down>下一页</a>  
        第".$page_id."页,共".$page_num."页";
             }
         $page_jump=$num_per_page*($page_id-1);
         $query=$query." limit $page_jump,$num_per_page";
        }
  }//end myfunction
?>
```

样式表文件代码省略。

程序清单　文档 head.php

```
<style type="text/css">
<!--
.STYLE1 {
    font-family:"华文彩云";
    font-size:36px;
}
-->
</style>
<table width="98%" border="0" align="center" cellpadding="0"
cellspacing="1">
  <tr>
<td height="60" bgcolor="5F8AC5"><span class="STYLE1">  
简易 BBS 系统</span></td>
  </tr>
  <tr>
    <td height="2"></td>
  </tr>
</table>
===================foot.php===================
<table width="98%" border="0" align="center" cellpadding="0"
cellspacing="0">
  <tr>
    <td height="10"></td>
  </tr>
  <tr>
    <td height="10" bgcolor="#5F8AC5"></td>
  </tr>
  <tr>
<td height="40" align="center" valign="middle">
建议使用浏览器 IE 6.0 以上 分辨率 1024*768 以上<br>
      版权所有：<a href="http://www.etpt.net" target="_blank">ET
```

```
 工作室</a> </td>
  </tr>
</table>
```

程序清单　文档 total.info.php

```
===================total_info.php===================
<?php
//用户登录并注册 SESSION
$tijiao=$_POST[tijiao];
if($tijiao=="提交"){
$user_name=$_POST[user_name];
$user_pw=$_POST[user_pw];
$check_query="select * from user_info where user_name='$user_name'";
$check_rst=$aa->excu($check_query);
$user=mysql_fetch_array($check_rst);
    if($user_pw==$user[user_pw]){
    $_SESSION[user_name]=$user[user_name];
    $today=date("Y-m-d H:i:s");
$query="update user_info set time2='$today' where
user_name='$_SESSION[user_name]'";
    $aa->excu($query);
    }
}
if($tijiao=="安全退出"){
$_SESSION[user_name]="";
}
?>
<table width="98%" border="0" align="center" cellpadding="0"
cellspacing="1">
  <tr><form id="form1" name="form1" method="post" action="#">
  <td width="80%" height="25" align="left" valign="middle"
bgcolor="5F8AC5"> 
    <?php
    if($_SESSION[user_name]!=""){
    echo "<font color=ffffff>欢迎您:".$_SESSION[user_name]."</font>";
    echo "         <input
 type=submit name=tijiao value=安全退出>";
    }else{
    ?>
       用户名:
       <input type="text" name="user_name" size="8" />
       密码:<input type="text" name="user_pw" size="8" />
       <input type="submit" name="tijiao" value="提交" />
```

```
      <a href="register.php"><font color="#FFFFFF">
我要注册</font></a>
    <?php
     }
     ?>
    </td>
     </form>
    <td width="20%" align="right" valign="middle" bgcolor="5F8AC5">
     <?php
      $today=date("Y-m-d H:i:s");
      echo $today;
     ?>
     </td>
   </tr>
   <tr>
<td height="25" colspan="2" align="right" valign="middle">贴子总数：<?php
echo $bb->note_total_num();?>  会员总数：<?php echo
$bb->user_total_num();?>  欢迎新会员：<?php echo
$bb->last_username();?>  </td>
   </tr>
   <tr>
    <td height="13" colspan="2" align="right" valign="middle"> </td>
   </tr>
 </table>
```

程序清单　文档 index.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>===简易 BBS 系统===</title>
<link href="inc/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<?php
include "inc/mysql.inc";
include "inc/myfunction.inc";
include "inc/head.php";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
include "inc/total_info.php";
?>
<table width="98%" border="0" align="center" cellpadding="0"
```

```
cellspacing="1" bgcolor="#FFFFFF">
  <tr>
<td width="50%" height="25" align="center" valign="middle"
bgcolor="5F8AC5">讨论区</td>
<td width="10%" align="center" valign="middle" bgcolor="5F8AC5">
主 题</td>
<td width="10%" align="center" valign="middle" bgcolor="5F8AC5">
帖 子</td>
<td width="20%" align="center" valign="middle" bgcolor="5F8AC5">
最新帖子</td>
<td width="10%" align="center" valign="middle" bgcolor="5F8AC5">
版 主</td>
  </tr>
  <tr>
    <td colspan="5">
     <?php
     $query="select * from father_module_info order by id";
     $result=$aa->excu($query);
     while($father_module=mysql_fetch_array($result)){
     ?>
     <table width="100%" border="0" cellspacing="0" cellpadding="0">
      <tr>
     <td height="25" colspan="6"
bgcolor="98B2CC">    <img src="pic_sys/li-1.gif"
width="16" height="15">    <?php echo
$father_module["module_name"]?></td>
      </tr>
      <?php
$query2="select * from son_module_info where
father_module_id='$father_module[id]' order by id";
      $result2=$aa->excu($query2);
      while($son_module=mysql_fetch_array($result2)){
      ?>
      <tr>
       <td width="5%" height="40" align="center" valign="middle"><img
   src="pic_sys/li-2.gif" width="32" height="32"></td>
       <td width="45%" align="left" valign="middle">
        <?php
        echo "<b><a href=module_list.php?module_id=
   ".$son_module[id]."><font
color=0000ff>".$son_module["module_name"]."</font></a></b><br>";
        echo $son_module["module_cont"];
        ?>      </td>
```

```
        <td width="10%" align="center" valign="middle"><?php echo
    $bb->son_module_idtonote_num($son_module["id"]);?></td>
        <td width="10%" align="center" valign="middle"><?php echo
    $bb->son_module_idtonote_num2($son_module["id"]);?></td>
        <td width="20%" align="left" valign="middle"><?php echo
    $bb->son_module_idtolast_note($son_module["id"]);?></td>
        <td width="10%" align="center" valign="middle"><?php echo
    $bb->son_module_idtouser_name($son_module["id"]);?></td>
      </tr>
      <?php }?>
    </table>
    <?php } ?>
    </td>
  </tr>
</table>
<?php
include "inc/foot.php";
?>
</body>
</html>
```

程序清单　文档 register.php

```
===================register.php===================
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>===简易 BBS 系统===</title>
<link href="inc/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<?php
include "inc/mysql.inc";
include "inc/myfunction.inc";
include "inc/head.php";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
include "inc/total_info.php";

?>
<table width="98%" border="0" align="center" cellpadding="0"
cellspacing="0">
  <tr>
```

```
<td width="73%" height="30"><a href="./">简易ＢＢＳ系统</a>>>
新用户注册</td>
<td width="27%" align="right" valign="middle">
<a href="new_note.php"></a></td>
  </tr>
 </table>
 <table width="98%" border="0" align="center" cellpadding="0"
 cellspacing="1" bgcolor="#FFFFFF">
  <tr>
<td height="25" align="center" valign="middle" bgcolor="5F8AC5">
发 布 新 贴</td>
  </tr>
  <tr>
    <td height="25" align="center" valign="middle">
     <?php
     //接收提交表单内容检验数据库中是否已经存在此用户名，不存在写入数据库
     $tijiao=$_POST[tijiao];
     if($tijiao=="提 交"){
         $user_name=$_POST[user_name];
         $query="select * from user_info where user_name='$user_name'";
         $rst=$aa->excu($query);
         if(mysql_num_rows($rst)!=0){
             echo "===您注册的用户名已经存在，请选择其他的用户名重新注册！===";
         }else{
             $user_pw1=$_POST[user_pw1];
             $user_pw2=$_POST[user_pw2];
             if($user_pw1!=$user_pw2){
                 echo "===您两次输入的密码不匹配，请重新输入！===";
             }else{
                 $today=date("Y-m-d H:i:s");
                 $query="insert into user_info(user_name,user_pw,time1)
             values('$user_name','$user_pw1','$today')";
                 if($aa->excu($query)){
                     echo "===恭喜您，注册成功！请<a href=../>返回主页</a>
                 登录===";
                     $register_tag=1;
                 }
             }
         }
     }
     //显示注册表单
     if($register_tag!=1){
     ?>
```

```
    <form name="form1" method="post" action="#">
    <table width="500" border="0" cellpadding="0" cellspacing="2">
     <tr>
       <td width="122" height="26" align="right" valign="middle"
  bgcolor="#CCCCCC">用户名:</td>
       <td width="372" height="26" align="left" valign="middle"
  bgcolor="#CCCCCC"><input type="text" name="user_name"></td>
     </tr>
     <tr>
       <td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
  密码:</td>
       <td height="26" align="left" valign="middle"
  bgcolor="#CCCCCC"><input type="text" name="user_pw1"></td>
     </tr>
     <tr>
       <td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
  重复密码:</td>
       <td height="26" align="left" valign="middle"
  bgcolor="#CCCCCC"><input type="text" name="user_pw2"></td>
     </tr>
     <tr>
<td height="26" colspan="2" align="center" valign="middle"
bgcolor="#CCCCCC"><input type="submit" name="tijiao" value="
提交">    <input type="reset" name="Submit2" value="
重置"></td>
       </tr>
    </table>
    </form>
    <?php
    }
    ?>
    </td>
  </tr>
  <tr>
    <td height="1" bgcolor="#CCCCCC"></td>
  </tr>
</table>
<?php
include "inc/foot.php";
?>
</body>
</html>
```

程序清单　文档 module_list.php

```
===================module_list.php===================
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>===简易 BBS 系统===</title>
<link href="inc/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<?php
include "inc/mysql.inc";
include "inc/myfunction.inc";
include "inc/head.php";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
include "inc/total_info.php";
$module_id=$_GET[module_id];
$del_id=$_GET[del_id];
if($bb->son_module_idtouser_name($module_id)==$_SESSION[user_name]){
    $query="delete from note_info where id='$del_id'";
    $aa->excu($query);
    }
$query="select * from note_info where module_id='$module_id' order by time
desc";
$add="module_id=".$module_id."&";
?>
<table width="98%" border="0" align="center" cellpadding="0"
cellspacing="0">
  <tr>
    <td width="73%" height="30"><a href="./">简易 B B S 系统</a>>>
    <?php
    echo "<a href=module_list.php?module_id=".$module_id.">";
    echo $bb->son_module_idtofather_name($module_id);
    echo "</a>>>";
    echo $bb->son_module_idtomodule_name($module_id);
    ?></td>
<td width="27%" align="right" valign="middle"><a
href="new_note.php"><img src="pic_sys/post.gif" width="82"
height="20"></a></td>
  </tr>
</table>
```

```
    <table width="98%" border="0" align="center" cellpadding="0"
    cellspacing="0">
      <tr>
    <td          height="30"          align="left"          valign="middle"><?php
$bb->page($query,$page_id,$add,20)?></td>
      </tr>
    </table>
    <table width="98%" border="0" align="center" cellpadding="0"
    cellspacing="1" bgcolor="#FFFFFF">
      <tr>
        <td     width="3%"     height="25"     align="center"     valign="middle"
bgcolor="5F8AC5"> </td>
    <td width="5%" align="center" valign="middle" bgcolor="5F8AC5">
    人气</td>
    <td width="50%" align="center" valign="middle" bgcolor="5F8AC5">
    标    题</td>
    <td width="9%" align="center" valign="middle" bgcolor="5F8AC5">
    发起人</td>
    <td width="5%" align="center" valign="middle" bgcolor="5F8AC5">
    帖子数</td>
    <td width="16%" align="center" valign="middle" bgcolor="5F8AC5">
    最后发表时间</td>
    <td width="12%" align="center" valign="middle" bgcolor="5F8AC5">
    最后发表人</td>
      </tr>
      <?php
        $result=$aa->excu($query);
        while($note=mysql_fetch_array($result)){
      ?>
      <tr>
    <td height="25" align="center" valign="middle"><img
    src="pic_sys/li-3.gif" width="14" height="11"></td>
    <td height="25" align="center" valign="middle"><?php echo
    $bb->note_idtotimes($note[id]);?></td>
        <td height="25" align="left" valign="middle"><?php
        echo "<a href=note_show.php?module_id=".$module_id."&note_id
    =".$note[id].">".$note[title]."</a>";
        if($bb->son_module_idtouser_name($module_id)==$_SESSION[user_name]
     || $_SESSION[manage_tag]==1){
        echo "  <a href=module_list.php?module_id=
    ".$module_id."&page_id=".$page_id."&del_id=".$note[id].">删除此帖</a>";
        }
        ?></td>
```

```
    <td height="25" align="center" valign="middle"><?php echo
$note[user_name];?></td>
  <td height="25" align="center" valign="middle"><?php echo
  $bb->note_idtonote_num($note[id]);?></td>
  <td height="25" align="center" valign="middle"><?php echo
  $bb->note_idtolast_time($note[id]);?></td>
  <td height="25" align="center" valign="middle"><?php echo
  $bb->note_idtolast_user_name($note[id]);?></td>
    </tr>
    <tr>
      <td height="1" colspan="7" bgcolor="#CCCCCC"></td>
    </tr>
    <?php
    }
    ?>
  </table>
  <table width="98%" border="0" align="center" cellpadding="0"
  cellspacing="0">
    <tr>
      <td height="30" align="right" valign="bottom">
       <?php
       $query="select * from note_info where module_id='$module_id' order by
  time desc";
       $bb->page($query,$page_id,$add,20)
       ?></td>
    </tr>
  </table>
  <?php
  include "inc/foot.php";
  ?>
  </body>
  </html>
```

程序清单　文档 new-note.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>===简易 BBS 系统===</title>
<link href="inc/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<?php
include "inc/mysql.inc";
```

```
include "inc/myfunction.inc";
include "inc/head.php";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
include "inc/total_info.php";
?>
<table width="98%" border="0" align="center" cellpadding="0"
cellspacing="0">
  <tr>
    <td width="73%" height="30"><a href="./">简易BBS系统</a>>>发新贴子</td>
<td width="27%" align="right" valign="middle"><a
href="new_note.php"><img src="pic_sys/post.gif" width="82"
height="20"></a></td>
  </tr>
</table>
<table width="98%" border="0" align="center" cellpadding="0"
cellspacing="1" bgcolor="#FFFFFF">
  <tr>
<td height="25" align="center" valign="middle" bgcolor="5F8AC5">
发 布 新 贴</td>
  </tr>
  <tr>
    <td height="25" align="center" valign="middle">
     <?php
     if($_SESSION[user_name]==""){
     echo "===请先登录！==";
     }else{
     //接收提交表单内容写入数据库
     $tijiao=$_POST[tijiao];
     if($tijiao=="提 交"){
         $module_id=$_POST[module_id];
         $title=$_POST[title];
         $cont=$_POST[cont];
         $cont=$bb->str_to($cont);
         $today=date("Y-m-d H:i:s");
         if($module_id!="" and $title!="" and $cont!=""){
             $query="insert into note_info(module_id,title,cont,time,user_
         name)values('$module_id','$title','$cont','$today',
         '$_SESSION[user_name]')";
             if($aa->excu($query)){
             echo "===新贴发布成功，请继续！===";
             }
```

```
    }else{
        echo "===请选择子模块，而且标题和内容均不能为空！===";
    }
}
?>
<form name="form1" method="post" action="">
<table width="500" border="0" cellpadding="0" cellspacing="2">
  <tr>
    <td width="122" height="26" align="right" valign="middle"
bgcolor="#CCCCCC">隶属模块:</td>
    <td width="372" height="26" align="left" valign="middle"
bgcolor="#CCCCCC">
     <?php
     $bb->son_module_list("");
     ?>
     </td>
  </tr>
  <tr>
    <td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
标题:</td>
    <td height="26" align="left" valign="middle"
bgcolor="#CCCCCC"><input type="text" name="title"></td>
  </tr>
  <tr>
    <td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
内容:</td>
    <td height="26" align="left" valign="middle"
bgcolor="#CCCCCC"><textarea name="cont" cols="50"
rows="8"></textarea></td>
  </tr>
  <tr>
    <td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
发贴人:</td>
    <td height="26" align="left" valign="middle"
bgcolor="#CCCCCC"> <?php echo $_SESSION[user_name];?></td>
  </tr>
  <tr>
    <td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
时间:</td>
    <td height="26" align="left" valign="middle" bgcolor="#CCCCCC">
系统将自动记录！</td>
  </tr>
  <tr>
```

```
        <td height="26" colspan="2" align="center" valign="middle"
    bgcolor="#CCCCCC"><input type="submit" name="tijiao" value="
    提 交">    <input type="reset" name="Submit2"
    value="重 置"></td>
        </tr>
     </table>
      </form>
      <?php
      }
      ?>

      </td>
  </tr>
  <tr>
    <td height="1" bgcolor="#CCCCCC"></td>
  </tr>
</table>
<?php
include "inc/foot.php";
?>
</body>
</html>
```

程序清单　文档 note_show.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>===简易 BBS 系统===</title>
<link href="inc/style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<?php
include "inc/mysql.inc";
include "inc/myfunction.inc";
include "inc/head.php";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
include "inc/total_info.php";
$module_id=$_GET[module_id];
$note_id=$_GET[note_id];
?>
<table width="98%" border="0" align="center" cellpadding="0"
```

```
cellspacing="0">
  <tr>
    <td width="73%" height="30"><a href="./">简易ＢＢＳ系统</a>>>
     <?php
     echo "<a href=module_list.php?module_id=".$module_id.">";
     echo $bb->son_module_idtofather_name($module_id);
     echo "</a>>>";
     echo $bb->son_module_idtomodule_name($module_id);
     //删除回复
     $del_id=$_GET[del_id];
     if($del_id!=""){

if($bb->son_module_idtouser_name($module_id)==$_SESSION[user_name]){
         $del_query="delete from note_info where id='$del_id'";
         $aa->excu($del_query);
         echo "<br>===删除回复成功！===";
         }
     }
     //添加回复
     $tijiao=$_POST[tijiao];
     if($tijiao=="提 交"){
         $title=$_POST[title];
         $cont=$_POST[cont];
         $cont=$bb->str_to($cont);
         $today=date("Y-m-d H:i:s");
         if($_SESSION[user_name]==""){
             $user_name="游客";
         }else{
             $user_name=$_SESSION[user_name];
         }
         $query="insert into note_info(up_id,title,cont,time,user_name)
    values('$note_id','$title','$cont','$today','$user_name')";
         $aa->excu($query);
     }
     ?></td>
<td width="16%" align="right" valign="middle"><a href="#huifu">
<img src="pic_sys/reply.gif" width="82" height="20"></a></td>
<td width="11%" align="right" valign="middle">
<a href="new_note.php"><img src="pic_sys/post.gif" width="82"
height="20"></a></td>
  </tr>
 </table>
 <?php
```

```
    $query="select * from note_info where up_id='$note_id' order by time";
    $add="module_id=".$module_id."&note_id=".$note_id."&";
    ?>
    <table width="98%" border="0" align="center" cellpadding="0"
cellspacing="0">
      <tr>
        <td height="30" align="left" valign="middle">
    <?php $bb->page($query,$page_id,$add,20)?></td>
      </tr>
    </table>
    <?php
    $query2="select * from note_info where id='$note_id'";
    $result2=$aa->excu($query2);
    $note2=mysql_fetch_array($result2);
    ?>

    <table width="98%" border="0" align="center" cellpadding="0"
    cellspacing="1" bgcolor="#FFFFFF">
      <tr>
    <td width="71%" height="25" align="left" valign="middle"
    bgcolor="5F8AC5">标题：<?php echo $note2[title]?></td>
    <td width="29%" align="center" valign="middle" bgcolor="5F8AC5">
    发帖时间：<?php echo $note2[time]?></td>
      </tr>
      <tr>
        <td height="1" colspan="2" bgcolor="#CCCCCC"></td>
      </tr>
    </table>
    <table width="98%" border="0" align="center" cellpadding="0"
    cellspacing="0">
      <tr>
    <td width="16%" height="15" align="center" valign="top">
    <img src="pic_sys/head2.jpg" width="85" height="90"><br><?php echo
    $note2[user_name]?></td>
        <td align="left" valign="middle"><?php echo $note2[cont]?></td>
      </tr>
      <tr>
    <td height="8" colspan="2" align="center" valign="top"
    bgcolor="#5F8AC5"></td>
      </tr>
    </table>
    <?php
    $rst=$aa->excu($query);
```

```
if(mysql_num_rows($rst)!=0){
?>
<table width="98%" border="0" align="center" cellpadding="0"
cellspacing="0">
  <?php
  while($note=mysql_fetch_array($rst)){
  ?>
  <tr>
    <td width="16%" height="120" rowspan="3" align="center" valign="top">
     <?php
     if($note[user_name]=="游客"){
     ?>
     <img src="pic_sys/head1.png" width="100" height="100">
     <br>
     游客
     <?php
     }else{
     ?>
     <img src="pic_sys/head2.jpg" width="85" height="90"><br>
     <?php
     echo $note[user_name];
     }
     ?>
     </td>
<td width="54%" height="26" align="left" valign="middle"><?php echo
$note[title]?></td>
<td width="18%" height="26" align="center" valign="middle"><?php echo
$note[time]?></td>
    <td width="12%" height="26" align="center" valign="middle">
     <?php
     if($bb->son_module_idtouser_name($module_id)==$_SESSION[user_name]
 || $_SESSION[manage_tag]==1){
     echo "<a href=?module_id=".$module_id."&note_id=
".$note_id."&page_id=".$page_id."&del_id=".$note[id].">删除回复</a>";
     }
     ?>  </td>
  </tr>
  <tr>
<td height="1" colspan="3" align="left" valign="top"
bgcolor="#CCCCCC"></td>
  </tr>

  <tr>
```

```
<td height="70" colspan="3" align="left" valign="top"><?php echo
$note[cont]?></td>
  </tr>
  <tr>
<td height="2" colspan="4" align="center" valign="top"
bgcolor="#CCCCCC"></td>
  </tr>
  <?php }?>
 </table>
 <?php
     }
 ?>
 <table width="98%" border="0" align="center" cellpadding="0"
 cellspacing="0">
  <tr>
    <td height="30" align="right" valign="bottom"><?php
     $query="select * from note_info where up_id='$note_id' order by time
 desc";
     $bb->page($query,$page_id,$add,20);
     ?></td>
  </tr>
 </table>
 <form name="form1" method="post" action="#">
 <table width="500" border="0" align="center" cellpadding="0"
 cellspacing="2">
  <tr>
<td height="26" colspan="2" align="center" valign="middle"
bgcolor="#CCCCCC"><a name="huifu">回 复 此 帖</a></td>
  </tr>
  <tr>
<td width="122" height="26" align="right" valign="middle"
bgcolor="#CCCCCC">标题:</td>
<td width="372" height="26" align="left" valign="middle"
bgcolor="#CCCCCC">

     <?php
        $reply_title="回复:".$bb->note_idtotitle($note_id);
        echo $reply_title;
     ?>
     <input type="hidden" name="title" value="<?php echo $reply_title?>">
     </td>
  </tr>
  <tr>
```

```
<td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
内容:</td>
<td height="26" align="left" valign="middle"
bgcolor="#CCCCCC"><textarea name="cont" cols="50"
rows="8"></textarea></td>
  </tr>
  <tr>
<td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
发贴人:</td>
    <td height="26" align="left" valign="middle" bgcolor="#CCCCCC"> 
     <?php
         if($_SESSION[user_name]==""){
             echo "游客";
         }else{
             echo $_SESSION[user_name];
         }
     ?></td>
  </tr>
  <tr>
<td height="26" align="right" valign="middle" bgcolor="#CCCCCC">
时间:</td>
<td height="26" align="left" valign="middle" bgcolor="#CCCCCC">
系统将自动记录! </td>
  </tr>
  <tr>
<td height="26" colspan="2" align="center" valign="middle"
bgcolor="#CCCCCC"><input type="submit" name="tijiao" value="提 交">
          
      <input type="reset" name="Submit2" value="重 置"></td>
  </tr>
 </table>
 </form>
 <?php
 include "inc/foot.php";
 ?>
 </body>
 </html>
```

程序清单　文档 login.php

```
 <?php
   include "../inc/mysql.inc";
    include "../inc/myfunction.inc";
    $aa=new mysql;
```

```
        $bb=new myfunction;
        $aa->link("");
        $_SESSION[manage_name]="";
        $_SESSION[manage_tag]="";
        $login=$_POST[login];
        if($login=="登录"){
            $username=$_POST[username];
            $password=$_POST[password];
            if($username!="" and password!=""){
                $query="select * from manage_user_info where
            user_name='$username'";
                $rst=$aa->excu($query);
                $user=mysql_fetch_array($rst,MySQL_ASSOC);
                if($user[user_pw]==$password){
                    $_SESSION[manage_name]=$user[user_name];
                    $_SESSION[manage_tag]=1;
                    header("location:./index.php");
                }
            }
        }
    ?>
    <head>
    <style>
    <!--
    td          { font-size:10pt }
    -->
    </style>
    <title>:::管理员登录==简易 BBS 系统:::</title>
    </head>
    <body background="../image/bg2.gif"
    onLoad="tijiao.username.value='';tijiao.username.focus();">

    <p align="center">  </p>
    <br>
      </font></p>
    <div align="center">
      <center>

      <form method=POST name="tijiao" action="#">
      <table border="1" cellpadding="0" cellspacing="0" bordercolor="#111111"
  width="240" height="126" bordercolorlight="#FFFFFF"
  bordercolordark="#FFFFFF" style="border-collapse:collapse">
        <tr>
```

```
    <td width="238" colspan="2" height="25" bgcolor="#A8A3AD">
    <p align="center"><b><font color="#FFFFFF">简易 BBS 后台管理系统
</font></b></td>
  </tr>
  <tr>
    <td width="64" height="26" bgcolor="#E3E1E6">
    <p align="center">账  号: </td>
    <td height="26" bgcolor="#E3E1E6" width="173">
    <input type="text" name="username" size="20" style="color:#A8A3AD;
border-style:solid; border-width:1; padding-left:4; padding-right:4;
padding-top:1; padding-bottom:1"></td>
  </tr>
  <tr>
    <td width="64" height="26" bgcolor="#E3E1E6">
    <p align="center">口  令: </td>
    <td height="26" bgcolor="#E3E1E6" width="173">
    <input type="password" name="password" size="20" style="color:
#A8A3AD; border-style:solid; border-width:1; padding-left:4;
padding-right:4; padding-top:1; padding-bottom:1"></td>
  </tr>
  <tr>
    <td width="238" height="29" bgcolor="#E3E1E6" colspan="2">
    <p align="center"><input type="submit" value="登 录"
name="login">    
    <input type="reset" value="取 消" name="cancel"></td>
  </tr>
  <tr>
    <td width="238" colspan="2" height="20" bgcolor="#A8A3AD">  </td>
  </tr>
 </table>
 </form>
 </center>
</div>
</body>
```

程序清单　文档 session.php

```
<?php
if($_SESSION[manage_name]=="" and $_SESSION[manage_tag]!=1){
   header("location:./login.php");
   }
?>
```

程序清单　文档 Admin_Style.css

此文件代码省略。

程序清单　文档 left.css

此文件代码省略。

程序清单　文档 menu.js

此文件代码省略。

程序清单　文档 index.php(manage 目录下的)

```
<?php
include "session.inc";
?>
<HTML>
<HEAD>
<TITLE>===简易 BBS 系统-后台管理===</TITLE>
<META http-equiv=Content-Type content="text/html; charset=gb2312">
</HEAD>

<FRAMESET id=frame border=false frameSpacing=0 rows=* frameBorder=0 
cols=200,* scrolling="yes">
   <FRAME name=left marginWidth=0 marginHeight=0 src="index_left.php" 
   scrolling=yes>
   <FRAMESET border=false frameSpacing=0 rows=53,* frameBorder=0 cols=* 
   scrolling="yes">
           <FRAME name=top src="index_top.php" scrolling=no>
           <FRAME name=main src="index_right.php">
      </FRAMESET>
</FRAMESET>
```

程序清单　文档 index_left.php

```
<?php
include "session.inc";
?>
<HTML>
<HEAD><TITLE>管理导航菜单</TITLE>
<META http-equiv=Content-Type content="text/html; charset=gb2312">
<SCRIPT src="menu.js"></SCRIPT>
<LINK href="left.css" type=text/css rel=stylesheet>
</HEAD>
<BODY leftMargin=0 topMargin=0 marginwidth="0" marginheight="0">
```

```
    <TABLE cellSpacing=0 cellPadding=0 width=180 align=center border=0>
      <TBODY>
      <TR>
    <TD vAlign=top height=44><IMG
    src="image/title.gif"></TD></TR></TBODY></TABLE>
    <TABLE cellSpacing=0 cellPadding=0 width=180 align=center>
      <TBODY>
      <TR>
        <TD class=menu_title id=menuTitle0
        onmouseover="this.className='menu_title2';"
        onmouseout="this.className='menu_title';"
        background=image/title_bg_quit.gif
          height=26>  <A href="Admin_Main.php"
      target="_top"><B><SPAN class=glow>管理首页</SPAN></B></A>
      <SPAN class=glow> |
          </SPAN><A  href="login.php"  target="_top"><B><SPAN  class=glow> 退 出
</SPAN></B></A> </TD></TR>
      <TR>
        <TD id=submenu0 background=image/title_bg_admin.gif height=97>
          <DIV style="WIDTH:180px">
          <TABLE cellSpacing=0 cellPadding=0 width=130 align=center>
            <TBODY>
            <TR>
      <TD height=16>您的用户名：<?php echo $_SESSION[manage_name];?></TD></TR>
            <TR>
       <TD height=16>您的身份：<?php echo $_SESSION[manage_name];?></TD></TR>
               <TR>
                <TD height=16>IP：<?php echo $_SERVER["REMOTE_ADDR"];?></TD>
              </TR>
            <TR>
             <TD height=16> </TD>
              </TR>
             </TBODY></TABLE></DIV>
          <DIV style="WIDTH:167px">
          <TABLE cellSpacing=0 cellPadding=0 width=130 align=center>
            <TBODY>
            <TR>
              <TD
height=20></TD></TR></TBODY></TABLE></DIV></TD></TR></TBODY></TABLE>
    <TABLE cellSpacing=0 cellPadding=0 width=167 align=center>
      <TBODY>
      <TR>
        <TD class=menu_title id=menuTitle1
```

```
        onmouseover="this.className='menu_title2'" style="CURSOR:hand"
        onclick="new Element.toggle('submenu1')"
        onmouseout="this.className='menu_title'"
    background=image/Admin_left_1.gif height=28 ;><SPAN class=glow>
    论坛模块管理</SPAN></TD>
      </TR>
      <TR>
        <TD id=submenu1 style="DISPLAY:none" align=right>
          <DIV class=sec_menu style="WIDTH:165px">
          <TABLE cellSpacing=0 cellPadding=0 width=132 align=center>
            <TBODY>
             <TR>
              <TD height=20><A href="father_module_add.php" target=main>
          父模块添加</A></TD>
             </TR>
             <TR>
      <TD height=20><A href="father_module_list.php" target=main>
  父模块管理</A></TD>
             </TR>
             <TR>
      <TD height=20><A href="son_module_add.php" target=main>
  子模块添加</A></TD>
             </TR>
            <TR>
      <TD height=20><A href="son_module_list.php" target=main>
  子模块管理</A></TD>
            </TR>
              </TABLE>
          </DIV>
          <DIV style="WIDTH:158px">
          <TABLE cellSpacing=0 cellPadding=0 width=130 align=center>
            <TBODY>
            <TR>
              <TD
height=4></TD></TR></TBODY></TABLE></DIV></TD></TR></TBODY></TABLE>
     <TABLE cellSpacing=0 cellPadding=0 width=167 align=center>
      <TBODY>
      <TR>
        <TD class=menu_title id=menuTitle2
        onmouseover="this.className='menu_title2'" style="CURSOR:hand"
        onclick="new Element.toggle('submenu2')"
        onmouseout="this.className='menu_title'"
    background=image/Admin_left_11.gif height=28 ;><SPAN class=glow>
```

```
论坛用户管理</SPAN></TD>
  </TR>
  <TR>
    <TD id=submenu2 style="DISPLAY:none" align=right>
      <DIV class=sec_menu style="WIDTH:165px">
      <TABLE cellSpacing=0 cellPadding=0 width=132 align=center>
        <TBODY>
        <TR>
          <TD height=20><A href="user_list.php" target=main>
      所有用户</A></TD>
        </TR>
         <TR>
          <TD height=20><A href="user_js.php" target=main>用户检索</A></TD>
         </TR>
 </TBODY></TABLE>
      </DIV>
      <DIV style="WIDTH:158px">
      <TABLE cellSpacing=0 cellPadding=0 width=130 align=center>
        <TBODY>
        <TR>
          <TD
height=4></TD></TR></TBODY></TABLE></DIV></TD></TR></TBODY></TABLE>
    <TABLE cellSpacing=0 cellPadding=0 width=167 align=center>
      <TBODY>
      <TR>
        <TD class=menu_title id=menuTitle3
        onmouseover="this.className='menu_title2'" style="CURSOR:hand"
        onclick="new Element.toggle('submenu3')"
        onmouseout="this.className='menu_title'"
    background=image/Admin_left_3.gif height=28 ;><SPAN class=glow>
    安全管理</SPAN></TD>
      </TR>
      <TR>
        <TD id=submenu3 style="DISPLAY:none" align=right>
          <DIV class=sec_menu style="WIDTH:165px">
          <TABLE cellSpacing=0 cellPadding=0 width=132 align=center>
            <TBODY>
             <TR>
    <TD height=20><A href="user_pw_change.php" target=main>
    密码更改</A></TD>
             </TR>
             <TR>
              <TD height=20><A href="../" target=_blank>帖子管理</A></TD>
```

```
        </TR>
    </TBODY></TABLE>
        </DIV>
        <DIV style="WIDTH:158px">
        <TABLE cellSpacing=0 cellPadding=0 width=130 align=center>
          <TBODY>
          <TR>
            <TD
height=4></TD></TR></TBODY></TABLE></DIV></TD></TR></TBODY></TABLE>
    <TABLE cellSpacing=0 cellPadding=0 width=167 align=center>
      <TBODY>
      <TR>
        <TD class=menu_title id=menuTitle208
        onmouseover="this.className='menu_title2';"
        onmouseout="this.className='menu_title';"
        background=image/Admin_left_04.gif
          height=28><SPAN>系统信息</SPAN> </TD></TR>
      <TR>
        <TD align=right>
          <DIV class=sec_menu style="WIDTH:165px">
          <TABLE cellSpacing=0 cellPadding=0 width=130 align=center>
            <TBODY>
            <TR>
              <TD height=20><br>版本号:verson 1.0<BR>版权所有： <A href=
          "http://www.etpt.net/" target=_blank>ET 工作室</A>
                <BR>设计制作： <A href="http://www.etpt.net/"
            target=_blank>ET 工作室</A>
                <BR>技术支持： <A href="http://bbs.etpt.net/"
            target=_blank>ET 工作室</A>

<BR><BR></TD></TR></TBODY></TABLE></DIV></TD></TR></TBODY></TABLE>
    </BODY>
    </HTML>
```

程序清单　文档 index_top.php

```
    <HTML>
    <HEAD><TITLE>顶部管理导航菜单</TITLE>
    <META http-equiv=Content-Type content="text/html; charset=gb2312">
    <STYLE type=text/css>A:link {
        COLOR:#ffffff; TEXT-DECORATION:none
    }
    A:hover {
        COLOR:#ffffff
```

```
    }
    A:visited {
        COLOR:#f0f0f0; TEXT-DECORATION:none
    }
    .spa {
        FONT-SIZE:9pt;
FILTER:Glow(Color=#0F42A6,Strength=2)dropshadow(Color=#0F42A6,OffX=2,OffY=1,
); COLOR:#8aade9; FONT-FAMILY:'宋体'
    }
    IMG {
        FILTER:Alpha(opacity:100); chroma:#FFFFFF)
    }
    </STYLE>
    <SCRIPT language=JavaScript type=text/JavaScript>
    function preloadImg(src){
      var img=new Image();
      img.src=src
    }
    preloadImg('image/admin_top_open.gif');

    var displayBar=true;
    function switchBar(obj){
      if(displayBar){
        parent.frame.cols='0,*';
        displayBar=false;
        obj.src='image/admin_top_open.gif';
        obj.title='打开左边管理导航菜单';
      } else {
        parent.frame.cols='200,*';
        displayBar=true;
        obj.src='image/admin_top_close.gif';
        obj.title='关闭左边管理导航菜单';
      }
    }
    </SCRIPT>
    <META content="MSHTML 6.00.2900.2963" name=GENERATOR></HEAD>
    <BODY leftMargin=0 background=image/admin_top_bg.gif
    topMargin=0>
    <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
      <TBODY>
      <TR vAlign=center>
        <TD width=60><IMG title=关闭左边管理导航菜单 style="CURSOR:hand"
          onclick=switchBar(this)
```

```
src="image/admin_top_close.gif"></TD>
    <TD width=92><A href="login.php" target="_parent" ><IMG
      src="image/top_an_1.gif" border=0></A></TD>
    <TD width=92><A href="#"></A></TD>
    <TD width=104><A href="#"></A></TD>
    <TD width=92><A href="#"></A></TD>
    <TD width=92><A href="#"></A></TD>
    <TD class=spa align=right>ET PHP SOUND CODE DEVELOP  
  </TD>
  </TR></TBODY></TABLE></BODY></HTML>
```

程序清单　文档 index_right.php

```
<?php
include "fun_head.php";
head("管理首页");
?>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD width=20 rowSpan=2> </TD>
    <TD class=topbg align=middle width=100>论坛模块管理</TD>
    <TD width=300> </TD>
    <TD width=40 rowSpan=2> </TD>
    <TD class=topbg align=middle width=100>论坛用户管理</TD>
    <TD width=300> </TD>
    <TD width=21 rowSpan=2> </TD></TR>
  <TR class=topbg2>
    <TD colSpan=2 height=1></TD>
    <TD colSpan=2></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD width=20> </TD>
    <TD width=400><br>
```

本管理某块共分四个子模块：父模块添加、父模块管理、子模块添加和子某块管理。其中父模块添加和子模块添加可以实现本论坛的父模块和子模块的添加功能，父模块的管理和子模块的管理可以实现本论坛父模块和子模块的删除、编辑等功能。

```
    <BR>
            </TD>
    <TD width=40> </TD>
    <TD
      width=400><br>
```

本管理某块共分两个子模块:所有用户和用户检索。其中所有用户按用户注册的后先顺序分页依次列出，用户检索是有管理员输入要查询的论坛注册用户的用户名，系统通过数据库查询出该用户的相关信息，并列出，而且管理员也可以在查询注册用户后直接删除该用户。

```
<br>
      <br></TD>
    <TD width=21> </TD></TR></TBODY></TABLE>
<TABLE height=10 cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD width=20 rowSpan=2> </TD>
    <TD class=topbg align=middle width=100>安全管理</TD>
    <TD width=300> </TD>
    <TD width=40 rowSpan=2> </TD>
    <TD class=topbg align=middle width=100>系统信息</TD>
    <TD width=300> </TD>
    <TD width=21 rowSpan=2> </TD></TR>
  <TR class=topbg2>
    <TD colSpan=2 height=1></TD>
    <TD colSpan=2></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD width=20> </TD>
    <TD
      width=400><br>
```

本管理模块共分两个子模块：密码更改和帖子管理。其中密码更改提供了管理员更改密码的功能，但必须输入原密码和两次新密码，帖子管理是直接进入论坛的主界面，但与普通注册用户所不同的是，在论坛的每个发帖和回复后面都多了一个删除按钮，可以通过此按钮删除相关帖子或回复。

```
</TD>
    <TD width=40> </TD>
    <TD
      width=400 valign="top"><br />
      本模块提供了本论坛的版本号、版权所有、设计制作以及技术支持等信息。</TD>
    <TD width=21> </TD></TR></TBODY></TABLE>
<TABLE height=70 cellSpacing=0 cellPadding=0 width="100%" border=0>
```

```
  <TBODY>
  <TR>
<TD></TD></TR></TBODY></TABLE>
<TABLE height=10 cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD></TD></TR></TBODY></TABLE>
<BR>
    <?php
    include "bottom.php";
    ?>
```

程序清单　文档 bottom.php

```
<TABLE class=border cellSpacing=1 cellPadding=2 width="100%" align=center
border=0>
  <TBODY>
  <TR align=middle>
    <TD class=topbg height=25><SPAN class=Glow>Copyright 2006 &copy; <a
href="http://www.etpt.net" target="_blank"><font color="#FFFFFF">ET 工 作 室
</font></a>
      All Rights Reserved.</SPAN> </TD></TR></TBODY></TABLE>
```

程序清单　文档 fun_head.php

```
<?php
 function head($str){
?>
<LINK href="Admin_Style.css" rel=stylesheet>
<STYLE type=text/css>
.STYLE4 {
    COLOR:#000000
}
</STYLE>
<BODY leftMargin=0 topMargin=0 marginheight="0" marginwidth="0">
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD width=392 rowSpan=2><img height=126
      src="image/adminmain01.gif" width=392></TD>
    <TD vAlign=top background=image/adminmain0line2.gif
    height=114>
      <TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
        <TBODY>
```

```
        <TR>
          <TD height=20></TD></TR>
        <TR>
          <TD><SPAN class=STYLE4>频道管理中心</SPAN></TD></TR>
        <TR>
          <TD height=8><IMG height=1
            src="image/adminmain0line.gif" width=283></TD></TR>
        <TR>
          <TD><IMG src="image/img_u.gif"
            align=absMiddle>欢迎进入管理<FONT
        color=#ff0000></FONT></TD></TR>
        <TR>
          <TD><IMG src="image/img_u.gif" align=absMiddle>
          <?php
          echo "当前位置：<b><font color=#ffffff>".$str."</font></b>";
          ?></TD>
        </TR>
        </TBODY></TABLE></TD></TR>
  <TR>
    <TD vAlign=bottom background=image/adminmain03.gif
    height=9><IMG height=12 src="image/adminmain02.gif"
      width=23></TD></TR></TBODY></TABLE>
      <?php
       }
?>
```

程序清单　文档 father_module_add.php

```
<?php
include "session.inc";
include "fun_head.php";
head("父模块添加");
include "../inc/mysql.inc";
include "../inc/myfunction.inc";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
$add_tag=$_GET[add_tag];
if($add_tag==1){
$show_order=$_POST[show_order];
$module_name=$_POST[module_name];
    if($show_order=="" or $module_name==""){
    echo "===对不起，您添加父模块不成功：<font color=red>
显示序号和父模块名称全不能为空</font>！===";
```

```
    }else{
    $query="insert into father_module_info
(module_name,show_order)values('$module_name','$show_order')";
    $aa->excu($query);
    echo "===恭喜您，添加父模块添加成功！===";
    }
}
?>
<table width="100%" height="389" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
  <tbody>
    <tr>
      <td width="20"> </td>
      <td valign="top"><br />
        <form action="?add_tag=1" method="post" name="form1" id="form1">
          <table width="408" height="87" border="0" align="center"
     cellpadding="0" cellspacing="1" bordercolor="#FFFFFF"
     bgcolor="449ae8">
            <tr bgcolor="#dcccccc">
              <td width="94" height="25" bgcolor="e0eef5"><div
          align="right">显示序号:</div></td>
              <td width="306" bgcolor="e0eef5"><input type="text" size="6"
          name="show_order" />
                  请填写一整数，如：1
              </td>
            </tr>
            <tr bgcolor="#dddddd">
              <td height="25" bgcolor="#FFFFFF"><div align="right">
          父模块名称:</div></td>
              <td bgcolor="#FFFFFF"><input type="text" size="20"
          name="module_name" /></td>
            </tr>
            <tr bgcolor="#dddddd">
              <td height="33" colspan="2" bgcolor="e0eef5">
          <div align="center">
                    <input name="submit" type="submit" value="提交" />
                        
                  <input name="reset" type="reset" value="重置" />
              </div></td>
            </tr>
          </table>
        </form>
        <br />
```

```
        <br /></td>
        <td width="20"> </td>
      </tr>
    </tbody>
  </table>
  <?php
      include "bottom.php";
  ?>
```

程序清单　文档 father_module_list.php

```
  <?php
  include "session.inc";
  include "fun_head.php";
  head("父模块管理");
  include "../inc/mysql.inc";
  include "../inc/myfunction.inc";
  $aa=new mysql;
  $bb=new myfunction;
  $aa->link("");
  ///////////删除父模块///////////////////////////////
  $del_tag=$_GET[del_tag];
  if($del_tag==1){
      $module_id=$_GET[module_id];
      $query="delete from father_module_info where id='$module_id'";
      $aa->excu($query);
      echo "==恭喜您，删除父模块信息成功！==<br>";
  }
  ///////////按显示顺序查询父模块信息表/////////////////////
      $query="select * from father_module_info order by show_order";
      $rst=$aa->excu($query);
  ?>
  <table width="100%" height="390" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
    <tbody>
      <tr>
        <td width="20"> </td>
        <td valign="top"><br /><table width="80%" border="0" align="center"
   cellpadding="0" cellspacing="1" bgcolor="449ae8">
          <tr bgcolor="#cccccc">
            <td width="92" height="23" bgcolor="e0eef5"><div align="center">
        编号</div></td>
            <td width="193" bgcolor="e0eef5"><div align="center">
        显示序号</div></td>
```

```
        <td width="368" bgcolor="e0eef5"><div align="center">
    父模块名称</div></td>
        <td colspan="2" bgcolor="e0eef5"><div align="center">
    操作</div></td>
      </tr>
<?php
    $m=0;
    while($module=mysql_fetch_array($rst,MySQL_ASSOC)){
    $m++;

?>
      <tr>
        <td height="19" bgcolor="#FFFFFF"><div align="center"><?php echo
     $m;?></div></td>
        <td bgcolor="#FFFFFF"><div align="center"><?php echo
    $module[show_order]?></div></td>
        <td bgcolor="#FFFFFF"><div align="center"><?php echo
    $module[module_name]?></div></td>
        <td width="134" align="center" bgcolor="#FFFFFF">
     <a href="father_module_bj.php?module_id=<?php echo
     $module[id];?>">编辑</a></td>
        <td width="142" align="center" bgcolor="#FFFFFF"><a
    href="?del_tag=1&module_id=<?php echo $module[id]?>">
    删除</a></td>
      </tr>
      <?php }?>
     </table>
     </td>
     <td width="20"> </td>
    </tr>
  </tbody>
</table>
  <?php
    include "bottom.php";
    ?>
```

程序清单　文档 father_module_bj.php

```
<?php
include "session.inc";
include "fun_head.php";
head("父模块管理==>>编辑");
include "../inc/mysql.inc";
include "../inc/myfunction.inc";
```

```
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
$module_id=$_GET[module_id];
$update_tag=$_GET[update_tag];
if($update_tag==1){
$show_order=$_POST[show_order];
$module_name=$_POST[module_name];
    if($show_order=="" or $module_name==""){
    echo "===对不起，您编辑父模块不成功：<font color=red>显示序号和父模块名称全
不能为空</font>！===";
    }else{
    $query="update father_module_info set
module_name='$module_name',show_order='$show_order' where
id='$module_id'";
    $aa->excu($query);
    echo "===恭喜您，编辑父模块添加成功！===";
    }
}
$query="select * from father_module_info where id='$module_id'";
$rst=$aa->excu($query);
$module=mysql_fetch_array($rst,MySQL_ASSOC);
?>
<table width="100%" height="389" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
    <tbody>
      <tr>
        <td width="20"> </td>
        <td valign="top"><br />
          <form action="?update_tag=1&module_id=<?php echo $module_id?>"
      method="post" name="form1" id="form1">
            <table width="408" height="87" border="0" align="center"
      cellpadding="0" cellspacing="1" bordercolor="#FFFFFF"
      bgcolor="449ae8">
              <tr bgcolor="#dcccccc">
                <td width="94" height="25" bgcolor="e0eef5"><div
            align="right">显示序号:</div></td>
                <td width="306" bgcolor="e0eef5"><input type="text" size="6"
            name="show_order" value="<?php echo $module[show_order]?>" />
                  请填写一整数，如：1
                </td>
              </tr>
              <tr bgcolor="#dddddd">
```

```
        <td height="25" bgcolor="#FFFFFF"><div align="right">
    父模块名称:</div></td>
        <td bgcolor="#FFFFFF"><input type="text" size="20"
     name="module_name" value="<?php echo $module[module_name]?>"
     /></td>
      </tr>
      <tr bgcolor="#dddddd">
        <td height="33" colspan="2" bgcolor="e0eef5"><div
    align="center">
            <input name="submit" type="submit" value="提交" />
                
          <input name="reset" type="reset" value="重置" />
        </div></td>
      </tr>
    </table>
  </form>
  <br />
<br /></td>
<td width="20"> </td>
</tr>
</tbody>
</table>
<?php
    include "bottom.php";
?>
```

程序清单 文档 son_module_add.php

```
<?php
include "session.inc";
include "fun_head.php";
head("子模块添加");
include "../inc/mysql.inc";
include "../inc/myfunction.inc";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
$add_tag=$_GET[add_tag];
if($add_tag==1){
$father_module_id=$_POST[father_module_id];
$module_name=$_POST[module_name];
$module_cont=$_POST[module_cont];
$user_name=$_POST[user_name];
    if($father_module_id=="" or $module_name=="" or $module_cont==""){
```

```
    echo "===对不起，您添加子模块不成功：<font color=red>隶属的父模块、子模块的名
称和简介全不能为空</font>！===";
    }else{
    $query="insert into son_module_info(father_module_id,module_name,
module_cont,user_name)values('$father_module_id','$module_name',
'$module_cont','$user_name')";
    $aa->excu($query);
    echo "===恭喜您，添加子模块添加成功！===";
    }
}
?>
<table width="100%" height="389" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
  <tbody>
    <tr>
      <td width="20"> </td>
      <td valign="top"><br />
        <form action="?add_tag=1" method="post" name="form1" id="form1">
          <table width="408" height="139" border="0" align="center"
     cellpadding="0" cellspacing="1" bordercolor="#FFFFFF"
     bgcolor="449ae8">
            <tr bgcolor="#dcccccc">
              <td width="94" height="25" bgcolor="e0eef5"><div
          align="right">隶属的父模块:</div></td>
              <td width="306" bgcolor="e0eef5"><?php
          $bb->father_module_list("");?></td>
            </tr>
            <tr bgcolor="#dddddd">
              <td height="25" bgcolor="#FFFFFF"><div align="right">
          子模块名称:</div></td>
              <td bgcolor="#FFFFFF"><input type="text" size="20"
          name="module_name" /></td>
            </tr>
            <tr bgcolor="#dddddd">
              <td height="25" align="right" valign="middle"
          bgcolor="#FFFFFF">简介:</td>
              <td bgcolor="#FFFFFF"><textarea name="module_cont" cols="42"
          rows="3"></textarea></td>
            </tr>
            <tr bgcolor="#dddddd">
              <td height="25" align="right" valign="middle"
         bgcolor="#FFFFFF">版主用户名:</td>
              <td bgcolor="#FFFFFF"><input type="text" size="20"
```

```
            name="user_name" />
                  可不填。</td>
              </tr>
              <tr bgcolor="#dddddd">
                <td height="33" colspan="2" bgcolor="e0eef5"><div
             align="center">
                    <input name="submit" type="submit" value="提交" />
                        
                  <input name="reset" type="reset" value="重置" />
                </div></td>
              </tr>
            </table>
          </form>
          <br />
        <br /></td>
        <td width="20"> </td>
      </tr>
    </tbody>
  </table>
  <?php
      include "bottom.php";
  ?>
```

程序清单　文档 son_module_list.php

```
<?php
include "session.inc";
include "fun_head.php";
head("子模块管理");
include "../inc/mysql.inc";
include "../inc/myfunction.inc";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
///////////删除子模块////////////////////////////////
$del_tag=$_GET[del_tag];
if($del_tag==1){
    $module_id=$_GET[module_id];
    $query="delete from son_module_info where id='$module_id'";
    $aa->excu($query);
    echo "==恭喜您，删除子模块信息成功！==<br>";
}
//////////////////按显示顺序查询父模块信息表////////////
    $query="select * from father_module_info order by show_order";
```

```
    $rst=$aa->excu($query);
?>
<table width="100%" height="390" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
  <tbody>
    <tr>
      <td width="20"> </td>
      <td valign="top"><br /><table width="80%" border="0" align="center"
 cellpadding="0" cellspacing="1" bgcolor="449ae8">
        <tr bgcolor="#cccccc">
          <td width="74" height="23" bgcolor="e0eef5"><div align="center">
    显示序号</div></td>
          <td width="84" bgcolor="e0eef5"><div align="center">
    父模块名称</div></td>
          <td width="410" bgcolor="e0eef5"><div align="center">
    子模块名称</div></td>
          <td colspan="2" bgcolor="e0eef5"><div align="center">
    操作</div></td>
        </tr>
        <?php
         while($father_module=mysql_fetch_array($rst,MySQL_ASSOC)){
         ?>
        <tr>
          <td height="19" bgcolor="#FFFFFF"><div align="center"><?php echo
    $father_module[show_order]?></div></td>
          <td colspan="4" align="left" valign="middle"
    bgcolor="#CCCCCC"><?php echo $father_module[module_name]?></td>
          </tr>
          <?php
////////从子模块信息表中按 id 顺序查询隶属该父模块的子模块的信息/////////////
          $query="select * from son_module_info where
    father_module_id='$father_module[id]' order by id";
          $rst2=$aa->excu($query);
          $m=0;
          while($son_module=mysql_fetch_array($rst2,MySQL_ASSOC)){
          $m++;
          ?>
        <tr>
          <td height="19" bgcolor="#FFFFFF"> </td>
          <td align="center" valign="middle" bgcolor="#FFFFFF"><?php echo
    $m?></td>
          <td bgcolor="#FFFFFF"><?php echo $son_module[module_name]?></td>
          <td width="80" align="center" bgcolor="#FFFFFF">
```

```
        <a href="son_module_bj.php?module_id=<?php echo
        $son_module[id];?>">编辑</a></td>
            <td width="80" align="center" bgcolor="#FFFFFF">
        <a href="?del_tag=1&module_id=<?php echo $son_module[id]?>">
        删除</a></td>
          <?php }?>
           </tr>
          <?php }?>
         </table>

         </td>
         <td width="20"> </td>
       </tr>
     </tbody>
   </table>
   <?php
       include "bottom.php";
   ?>
```

程序清单　文档 son_module_bj.php

```
   <?php
   include "session.inc";
   include "fun_head.php";
   head("子模块管理>>编辑");
   include "../inc/mysql.inc";
   include "../inc/myfunction.inc";
   $aa=new mysql;
   $bb=new myfunction;
   $aa->link("");
   $module_id=$_GET[module_id];
   $update_tag=$_GET[update_tag];
   if($update_tag==1){
   $father_module_id=$_POST[father_module_id];
   $module_name=$_POST[module_name];
   $module_cont=$_POST[module_cont];
   $user_name=$_POST[user_name];
       if($father_module_id=="" or $module_name=="" or $module_cont==""){
       echo "===对不起，您编辑子模块不成功：<font color=red>隶属的父模块、子模块的
   名称和简介全不能为空</font>! ===";
       }else{
       $query="update son_module_info set
  father_module_id='$father_module_id',module_name='$module_name',
  module_cont='$module_cont',user_name='$user_name' where id=
```

```
'$module_id'";
     $aa->excu($query);
     echo "===恭喜您，编辑子模块添加成功！===";
     }
}
$query="select * from son_module_info where id='$module_id'";
$rst=$aa->excu($query);
$module=mysql_fetch_array($rst,MySQL_ASSOC);
?>
<table width="100%" height="389" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
  <tbody>
    <tr>
      <td width="20"> </td>
      <td valign="top"><br />
        <form action="?update_tag=1&module_id=<?php echo $module_id?>"
   method="post" name="form1" id="form1">
          <table width="408" height="139" border="0" align="center"
     cellpadding="0" cellspacing="1" bordercolor="#FFFFFF"
     bgcolor="449ae8">
            <tr bgcolor="#dcccccc">
              <td width="94" height="25" bgcolor="e0eef5">
         <div align="right">隶属的父模块:</div></td>
              <td width="306" bgcolor="e0eef5"><?php
         $bb->father_module_list($module[father_module_id]);?></td>
            </tr>
            <tr bgcolor="#dddddd">
              <td height="25" bgcolor="#FFFFFF"><div align="right">
         子模块名称:</div></td>
              <td bgcolor="#FFFFFF"><input type="text" size="20"
         name="module_name" value="<?php echo $module[module_name]?>"
         /></td>
            </tr>
            <tr bgcolor="#dddddd">
              <td height="25" align="right" valign="middle"
         bgcolor="#FFFFFF">简介:</td>
              <td bgcolor="#FFFFFF"><textarea name="module_cont" cols="42"
         rows="3"><?php echo $module[module_cont]?></textarea></td>
            </tr>
            <tr bgcolor="#dddddd">
              <td height="25" align="right" valign="middle"
          bgcolor="#FFFFFF">版主用户名:</td>
              <td bgcolor="#FFFFFF"><input type="text" size="20"
```

```
            name="user_name" value="<?php echo $module[user_name]?>" />
                  可不填。</td>
              </tr>
              <tr bgcolor="#dddddd">
                <td height="33" colspan="2" bgcolor="e0eef5">
            <div align="center">
                    <input name="submit" type="submit" value="提交" />
                        
                  <input name="reset" type="reset" value="重置" />
                </div></td>
              </tr>
            </table>
          </form>
          <br />
        <br /></td>
        <td width="20"> </td>
      </tr>
    </tbody>
  </table>
  <?php
      include "bottom.php";
  ?>
```

程序清单　文档 user_list.php

```
<?php
include "session.inc";
include "fun_head.php";
head("所有用户");
include "../inc/mysql.inc";
include "../inc/myfunction.inc";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
///////////删除注册用户////////////////////////////////
$del_tag=$_GET[del_tag];
if($del_tag==1){
    $user_id=$_GET[user_id];
    $query="delete from user_info where id='$user_id'";
    $aa->excu($query);
    echo "==恭喜您，删除注册用户信息成功！==<br>";
}
/////////////从用户信息表中查询所有用户/////////////////
    $query="select * from user_info order by id desc";
```

```
?>
<table width="100%" height="390" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
  <tbody>
    <tr>
      <td width="20"> </td>
      <td valign="top"><br />
        <table width="80%" border="0" align="center" cellpadding="0"
   cellspacing="0">
          <tr>
            <td      height="30"      align="left"      valign="middle"><?php
$bb->page($query,$page_id,$add,20)?></td>
          </tr>
        </table>
        <table width="80%" border="0" align="center" cellpadding="0"
   cellspacing="1" bgcolor="449ae8">
        <tr bgcolor="#cccccc">
          <td width="46" height="23" bgcolor="e0eef5"><div align="center">
     序号</div></td>
          <td width="213" bgcolor="e0eef5"><div align="center">
     用户名</div></td>
          <td width="201" bgcolor="e0eef5"><div align="center">
     注册时间</div></td>
          <td width="188" bgcolor="e0eef5"><div align="center">
     最后登录时间</div></td>
          <td width="80" bgcolor="e0eef5"><div align="center">
     操作</div></td>
        </tr>
<?php
    $rst=$aa->excu($query);
    $m=0;
    while($user=mysql_fetch_array($rst,MySQL_ASSOC)){
    $m++;
?>
        <tr>
          <td height="19" bgcolor="#FFFFFF"><div align="center"><?php echo
     $m;?></div></td>
          <td bgcolor="#FFFFFF"><div align="center"><?php echo
     $user[user_name]?></div></td>
          <td bgcolor="#FFFFFF"><div align="center"><?php echo
     $user[time1]?></div></td>
          <td align="center" bgcolor="#FFFFFF"><?php echo
     $user[time2]?></td>
```

```
          <td align="center" bgcolor="#FFFFFF">
      <a href="?del_tag=1&user_id=<?php echo $user[id]?>">删除</a></td>
          </tr>
        <?php }?>
      </table>
        <table width="80%" border="0" align="center" cellpadding="0"
    cellspacing="0">
          <tr>
            <td height="30" align="right" valign="middle">
              <?php
              $query="select * from user_info order by id desc";
              $bb->page($query,$page_id,$add,20);
              ?></td>
          </tr>
        </table></td>
      <td width="20"> </td>
    </tr>
  </tbody>
</table>
<?php
    include "bottom.php";
?>
```

程序清单　文档 user_js.php

```
<?php
include "session.inc";
include "fun_head.php";
head("用户检索");
include "../inc/mysql.inc";
include "../inc/myfunction.inc";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
//////////删除用户//////////////////////////////
$del_tag=$_GET[del_tag];
if($del_tag==1){
    $del_id=$_GET[del_id];
    $query="delete from user_info where id='$del_id'";
    $aa->excu($query);
    echo "==恭喜您，删除用户信息成功，请继续！==<br>";
}
//////////////////////////////////////////////////
    $user_name=$_POST[user_name];
```

```
?>
<table width="100%" height="390" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
  <tbody>
    <tr>
      <td width="20"> </td>
      <td valign="top"><br />
        <table width="70%" border="0" align="center" cellpadding="0"
   cellspacing="1" bgcolor="449ae8">
        <tr bgcolor="#cccccc">
         <form id="form1" name="form1" method="post" action="?">
           <td height="23" bgcolor="e0eef5"><div align="center">
             <input type="text" name="user_name" size="16" value="<?php echo
         $user_name?>" />    
             <input type="submit" name="Submit" value="提交" />
            </div></td>
          </form>
           </tr>
          <tr>
          <td height="19" align="center" bgcolor="#FFFFFF">
           <?php
           if($user_name==""){
             echo "请输入您要检索的用户名，点提交查询！";
           }else{
             $query="select * from user_info where user_name='$user_name'";
             $rst=$aa->excu($query);
             if(mysql_num_rows($rst)==0){
                 echo "很抱歉，系统没有检索到您要查找的用户！";
             }else{
                 $user=mysql_fetch_array($rst,MySQL_ASSOC);
           ?>
           <table width="100%" border="0" cellspacing="2" cellpadding="0">
            <tr>
             <td width="29%" height="20" align="right" bgcolor="#dddddd">
         用户名:</td>
             <td width="71%" align="left" bgcolor="#dddddd"><?php echo
         $user[user_name]?></td>
            </tr>
            <tr>
             <td height="20" align="right" bgcolor="#dddddd">登录口令:</td>
             <td align="left" bgcolor="#dddddd"><?php echo
        $user[user_pw]?></td>
            </tr>
```

```
              <tr>
                <td height="20" align="right" bgcolor="#dddddd">注册时间:</td>
                <td align="left" bgcolor="#dddddd"><?php echo
            $user[time1]?></td>
              </tr>
              <tr>
                <td height="20" align="right" bgcolor="#dddddd">
            最后登录时间:</td>
                <td align="left" bgcolor="#dddddd"><?php echo
            $user[time2]?></td>
              </tr>
              <tr>
                <td height="20" colspan="2" align="center"
            bgcolor="#dddddd"><a href=?del_tag=1&del_id=<?php echo
            $user[id]?>>删除此用户</a></td>
                </tr>
             </table>
              <?php
                }
              }
              ?>
              </td>
             </tr>
        </table>
        </td>
        <td width="20"> </td>
      </tr>
    </tbody>
  </table>
    <?php
      include "bottom.php";
    ?>
```

程序清单　文档 user_pw_change.php

```
<?php
include "session.inc";
include "fun_head.php";
head("密码更改");
include "../inc/mysql.inc";
include "../inc/myfunction.inc";
$aa=new mysql;
$bb=new myfunction;
$aa->link("");
```

```
///////////更改密码///////////////////////////////
$tijiao=$_POST[tijiao];
if($tijiao=="提 交"){
    $pw_old=$_POST[pw_old];
    $pw_new1=$_POST[pw_new1];
    $pw_new2=$_POST[pw_new2];
    if($pw_new1!=$pw_new2){
        echo "===您两次输入的新密码不匹配,请重新输入!===";
    }else{
        $query="select * from manage_user_info where
    user_name='$_SESSION[manage_name]' and user_pw='$pw_old'";

        $rst=$aa->excu($query);
        if(mysql_num_rows($rst)==0){
            echo "===您输入的旧密码不正确,请重新输入!===";
        }else{
            $query="update manage_user_info set user_pw='$pw_new1' where
        user_name='$_SESSION[manage_name]'";
            $aa->excu($query);
            echo "===恭喜您,您的登录密码修改成功!===";
        }
    }
}
?>
<table width="100%" height="390" border="0" cellpadding="0" cellspacing="0"
bgcolor="f0f0f0">
      <tbody>
        <tr>
          <td width="20"> </td>
          <td valign="top"><br />
            <form id="form1" name="form1" method="post" action="#">
            <table width="60%" border="0" align="center" cellpadding="0"
       cellspacing="1" bgcolor="449ae8">
            <tr bgcolor="#cccccc">
               <td width="27%" height="23" align="right" bgcolor="e0eef5">
          用户名:</td>
               <td width="73%" align="left" bgcolor="e0eef5"><?php echo
          $_SESSION[manage_name]?></td>
            </tr>
              <tr>
              <td height="19" align="right" bgcolor="#FFFFFF">原口令:</td>
              <td align="left" bgcolor="#FFFFFF"><input type="text"
         name="pw_old" size="16" /></td>
```

```
          </tr>
          <tr>
            <td height="19" align="right" bgcolor="#FFFFFF">新密码:</td>
            <td align="left" bgcolor="#FFFFFF"><input type="text"
        name="pw_new1" size="16" /></td>
          </tr>
          <tr>
            <td height="19" align="right" bgcolor="#FFFFFF">再次新密码:</td>
            <td align="left" bgcolor="#FFFFFF"><input type="text"
        name="pw_new2" size="16" /></td>
          </tr>
          <tr>
            <td height="19" colspan="2" align="center"
        bgcolor="#FFFFFF"><input type="submit" name="tijiao" value="
        提 交" />
               
              <input type="reset" name="Submit2" value="重 置" /></td>
          </tr>
        </table>
        </form>
        </td>
        <td width="20"> </td>
      </tr>
    </tbody>
  </table>
    <?php
      include "bottom.php";
    ?>
```

程序清单 文档 install.php

```
<?php
include "inc/mysql.inc";
include "inc/myfunction.inc";
$aa=new mysql;
$bb=new mysql;
$aa->link("mysql");
//创建数据库
$query="CREATE DATABASE `bbs_data`";
if($aa->excu($query)){
    echo "===数据库创建成功! ===";
    }
$bb->link("bbs_data");
```

```
//////////创建数据表 father_module_info////////////
$query="CREATE TABLE 'father_module_info' (
  'id' int(11)NOT NULL auto_increment,
  'module_name' varchar(66)default NULL,
  'show_order' int(11)default '0',
  UNIQUE KEY 'id' ('id')
)";
$bb->excu($query);
echo "<br>  数据表-father_module_info-创建成功！";
//////////创建数据表 manage_user_info////////////
$query="CREATE TABLE 'manage_user_info' (
  'id' int(11)NOT NULL auto_increment,
  'user_name' varchar(16)NOT NULL,
  'user_pw' varchar(16)NOT NULL,
  UNIQUE KEY 'id' ('id')
)";
$bb->excu($query);
echo "<br>  数据表-manage_user_info-创建成功！";
//////////创建数据表 note_info////////////
$query="CREATE TABLE 'note_info' (
  'id' int(11)NOT NULL auto_increment,
  'module_id' int(11)default '0',
  'up_id' int(11)default '0',
  'title' varchar(88)NOT NULL,
  'cont' text NOT NULL,
  'time' datetime NOT NULL,
  'user_name' varchar(16)NOT NULL,
  'times' int(11)default '0',
  UNIQUE KEY 'id' ('id')
)";
$bb->excu($query);
echo "<br>  数据表-note_info-创建成功！";
//////////创建数据表 son_module_info////////////
$query="CREATE TABLE 'son_module_info' (
  'id' int(11)NOT NULL auto_increment,
  'father_module_id' int(11)NOT NULL,
  'module_name' varchar(66)NOT NULL,
  'module_cont' text NOT NULL,
  'user_name' varchar(16)default NULL,
  UNIQUE KEY 'id' ('id')
```

```
)";
$bb->excu($query);
echo "<br>  数据表-son_module_info-创建成功！";
//////////创建数据表 user_info////////////
$query="CREATE TABLE 'user_info' (
  'id' int(11)NOT NULL auto_increment,
  'user_name' varchar(16)NOT NULL,
  'user_pw' varchar(16)NOT NULL,
  'time1' datetime default '0000-00-00 00:00:00',
  'time2' datetime default '0000-00-00 00:00:00',
  UNIQUE KEY 'id' ('id')
)";
$bb->excu($query);
echo "<br>  数据表-user_info-创建成功！";
//////////////////初始化数据/////////////////////////
echo "<br>===初始化基本数据===";
//////////////////初始化管理用户名和密码/////////////////////////
$query="INSERT INTO `manage_user_info` VALUES(1,'admin','admin')";
$aa->excu($query);
echo "<br>  管理员用户名、密码初始成功";
echo "<br>   用户名：admin";
echo "<br>   密 码：admin";
//////////////////初始化测试父模块/////////////////////////
$query="INSERT INTO 'father_module_info' VALUES(1,'测试父模块',1)";
$aa->excu($query);
echo "<br>  添加"测试父模块"成功！";
//////////////////初始化测试子模块/////////////////////////
$query="INSERT INTO 'son_module_info' VALUES(1,1,'测试子模块',
'这是为了测试子模块而添加的','');";
$aa->excu($query);
echo "<br>  添加"测试子模块"成功！";
?>
```

10.5 程 序 运 行

（1）执行安装程序，执行文件的地址和效果如图 10-3 所示。

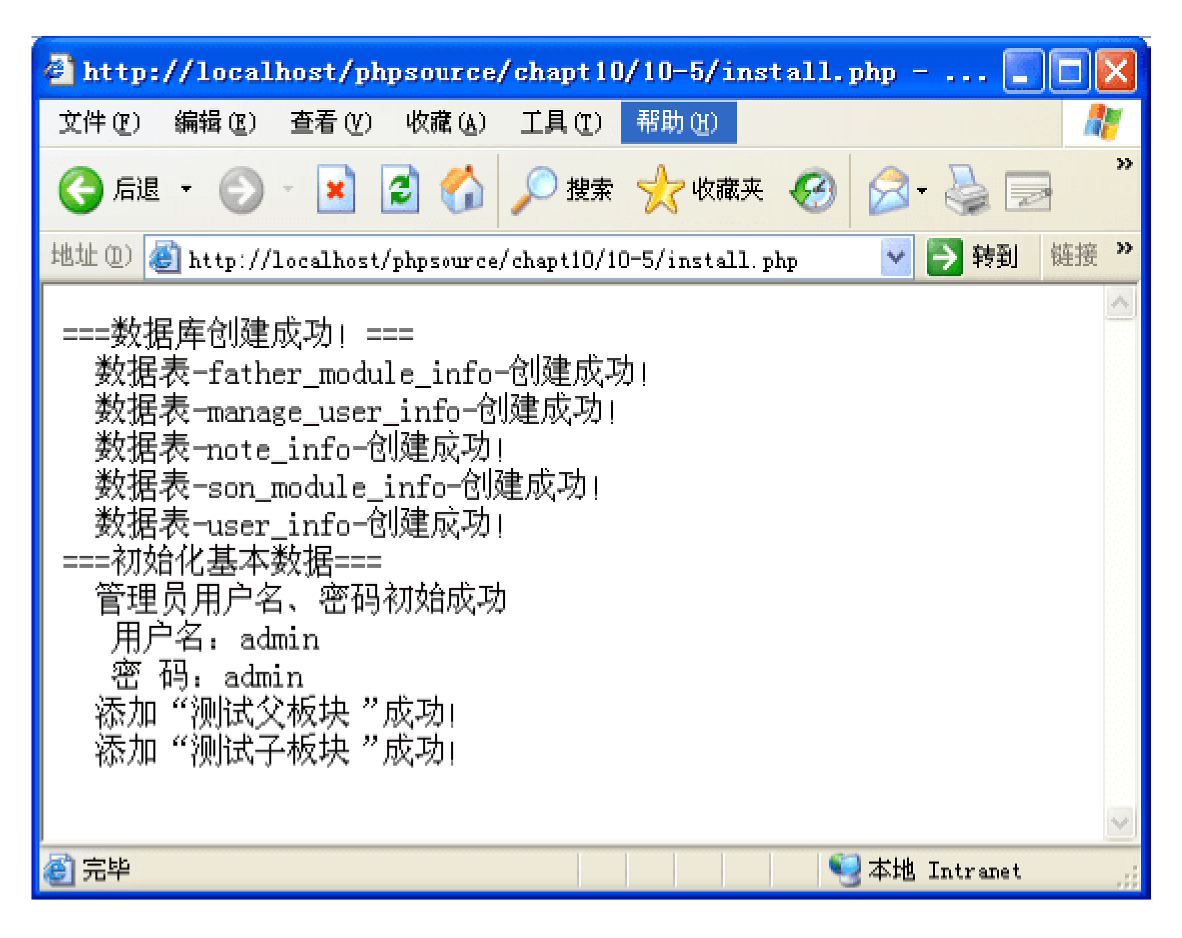


图 10-3　安装程序页面

（2）进入论坛首页，执行文件的地址和效果如图 10-4 所示。

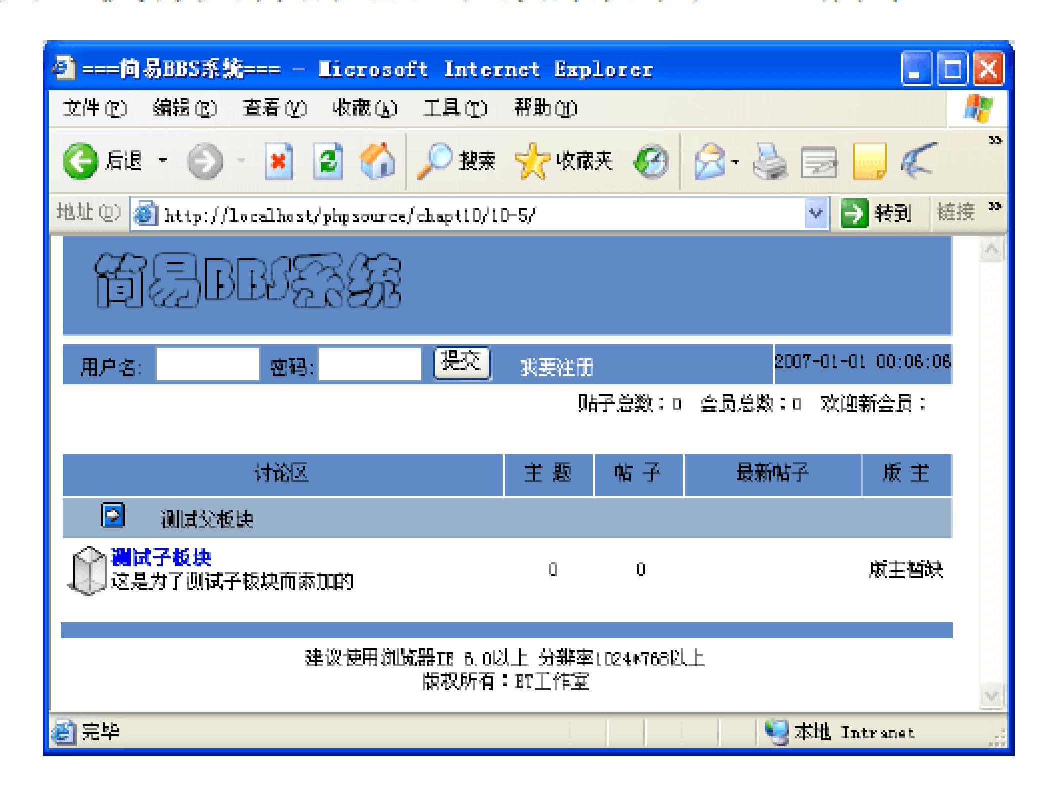


图 10-4　论坛首页

（3）注册新用户。单击图 10-4 中“我要注册”超级链接，其运行效果图如图 10-5 所示。

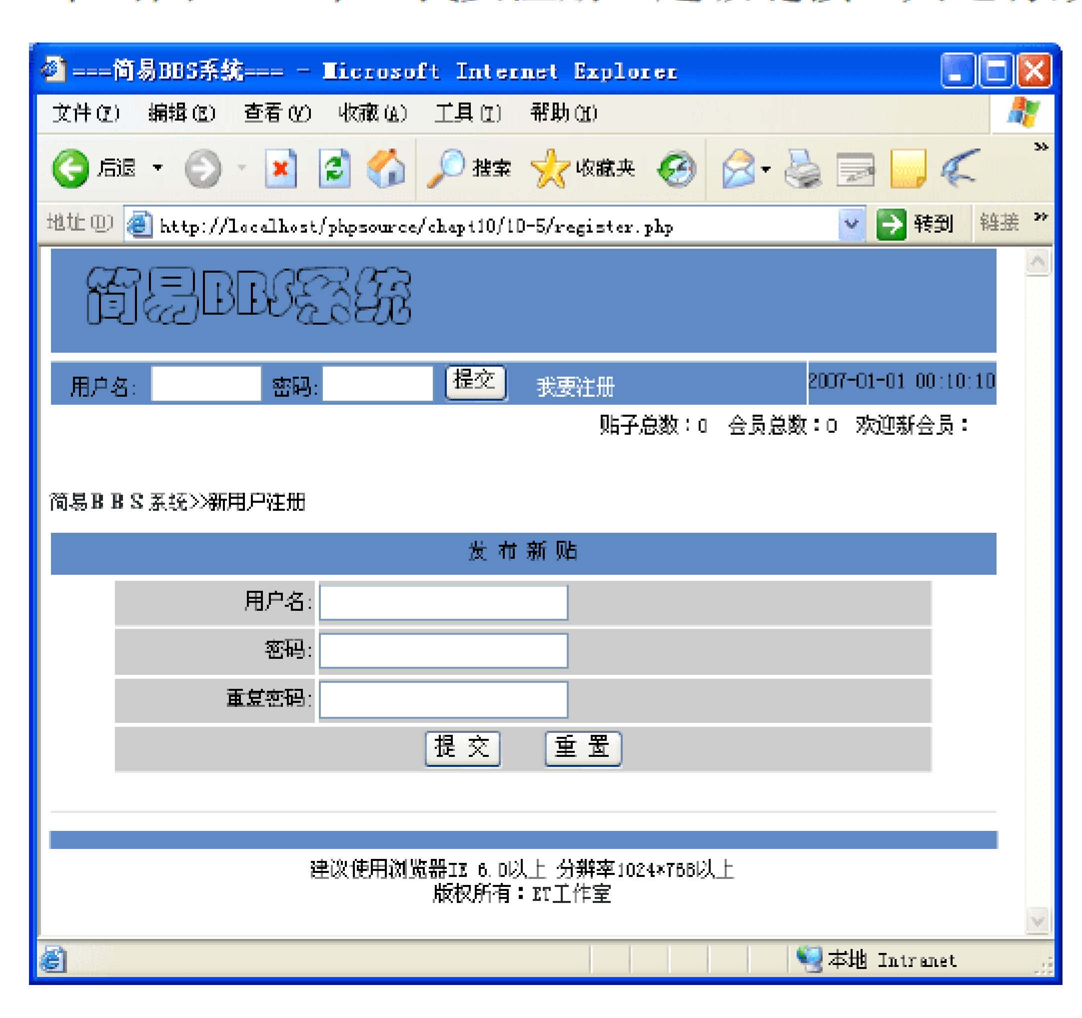


图 10-5　注册新用户页面

注册成功的效果图如图 10-6 所示。



图 10-6　注册成功页面

（4）返回主页登录。单击图 10-6 中的“返回主页”超级链接，登录成功的运行效果图如图 10-7 所示。

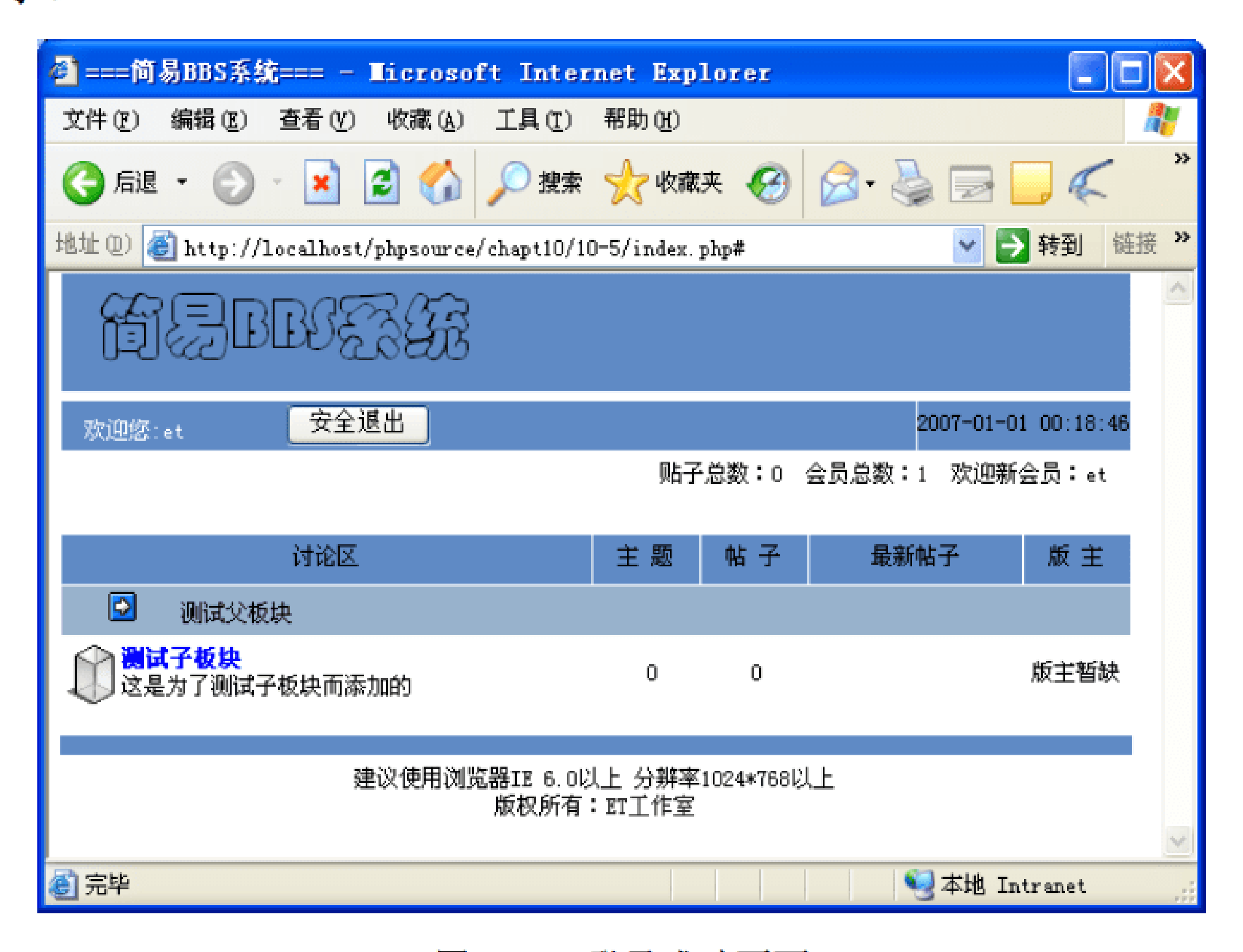


图 10-7　登录成功页面

（5）查看“测试子模块”中的内容。单击图 10-7 的“测试父模块”下面的“测试子模块”超级链接，其效果如图 10-8 所示。

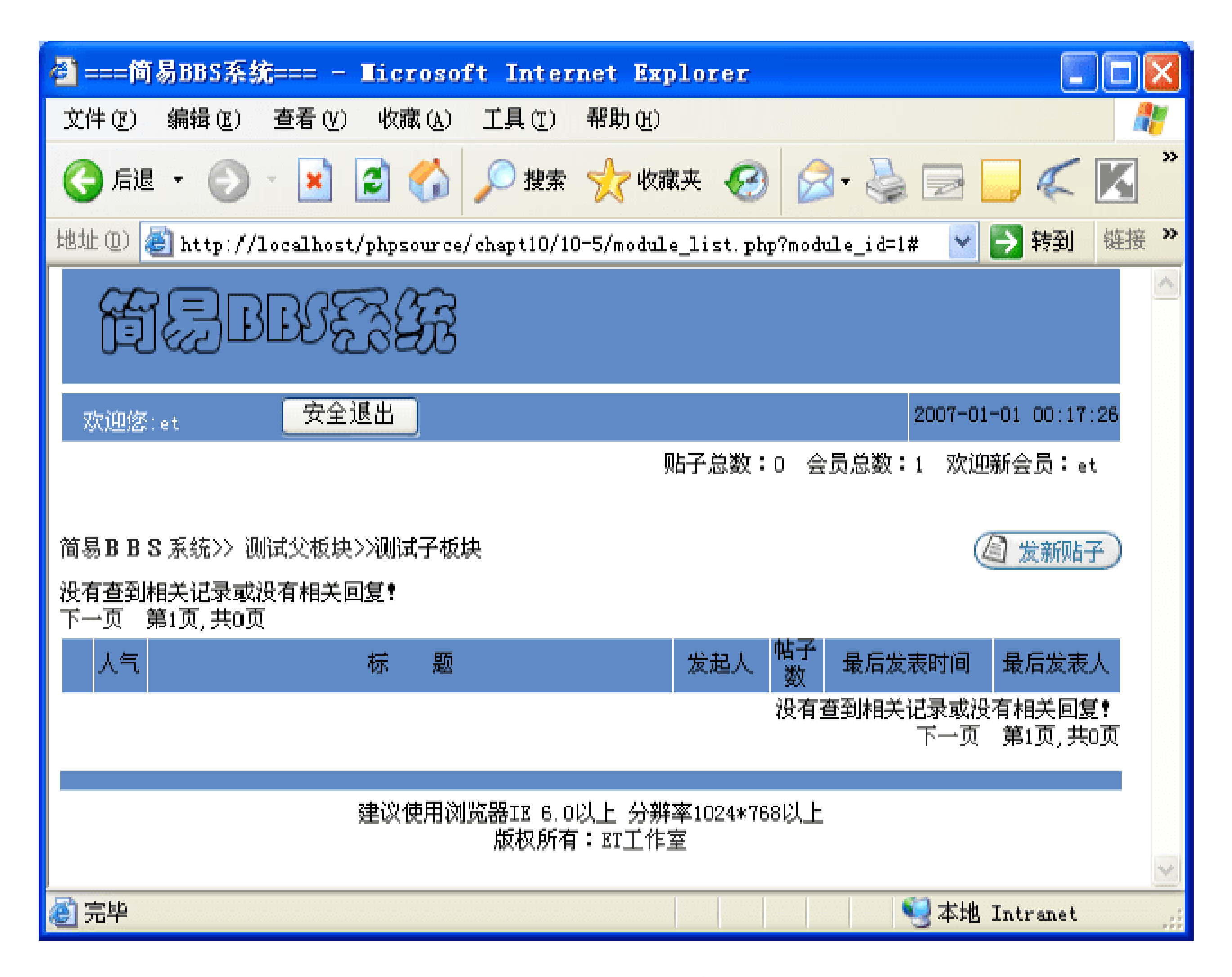


图 10-8　测试子模块页面

因没有发布帖子所以没有内容可以显示。

（6）发布新帖。单击图 10-8 的“发新帖子”按钮，其运行效果如图 10-9 所示。

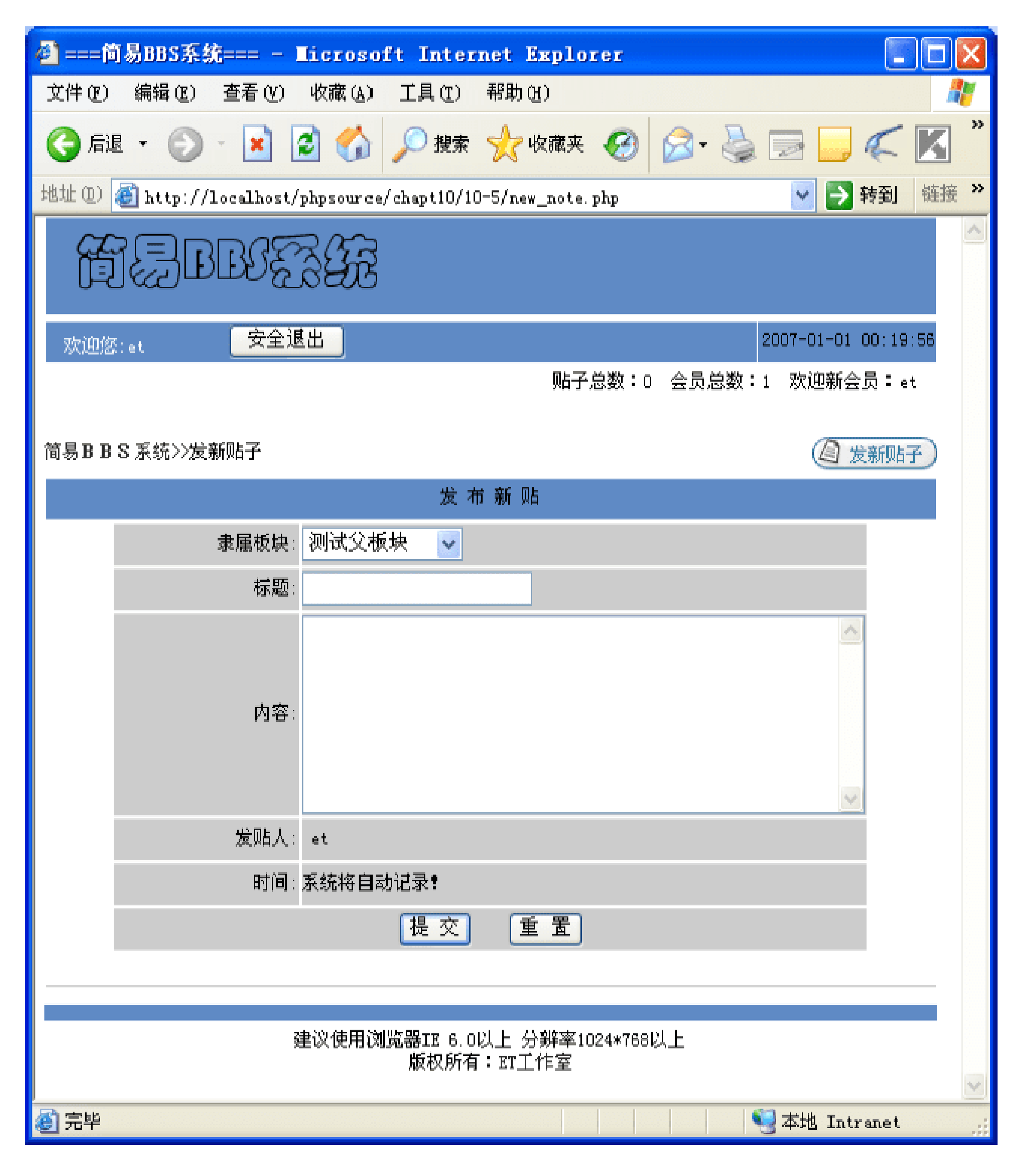


图 10-9　发新帖子页面

填写图 10-9 的表单，单击“提交”按钮，发布成功出现“===新贴发布成功，请继续！===”字样。

（7）再次查看“测试子模块”中的内容。同步骤 5。其效果如图 10-10 所示。

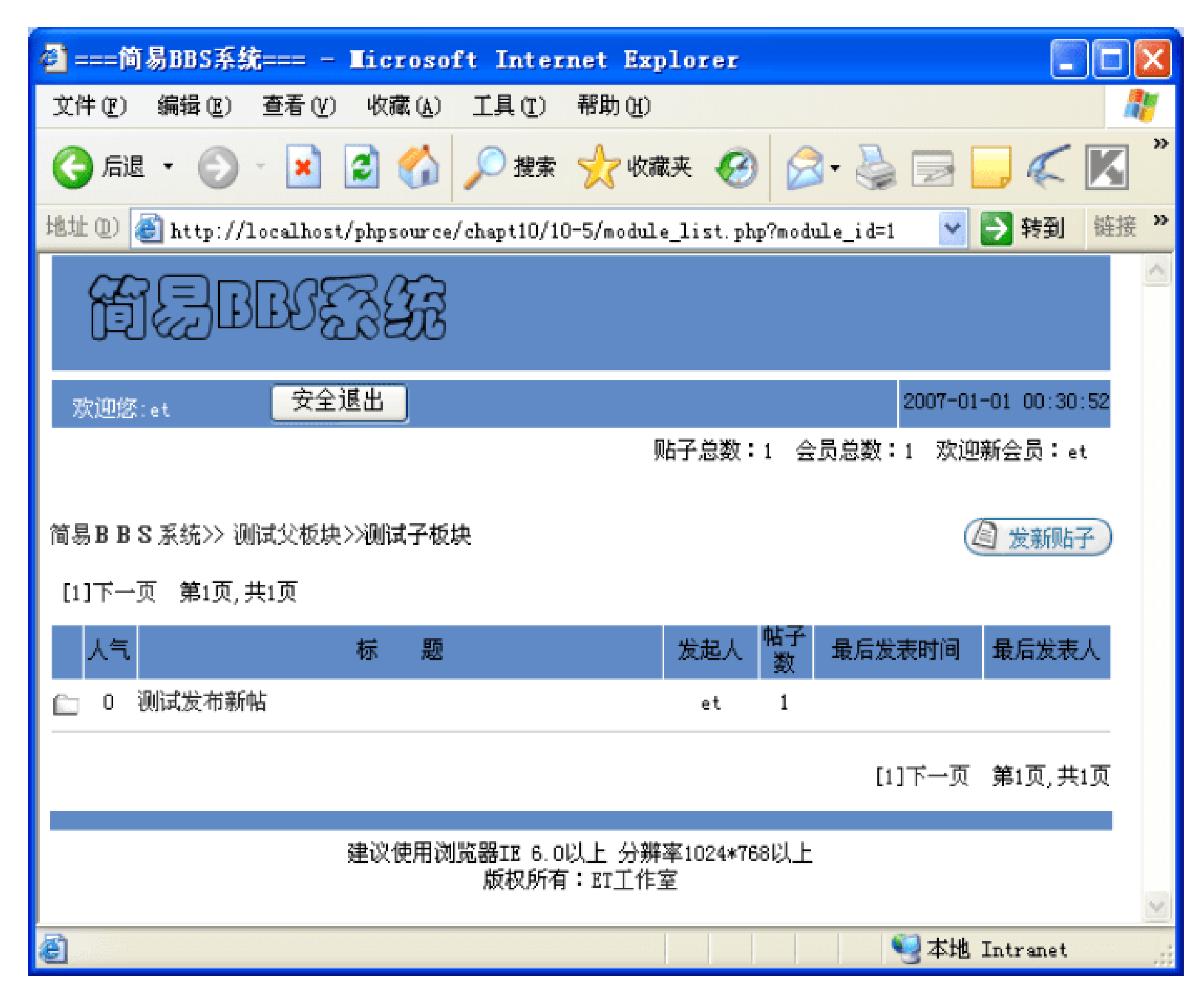


图 10-10 测试子模块页面

（8）查看帖子的具体内容。单击图 10-10 的“测试发布新帖”超级链接，其运行效果如图 10-11 所示。

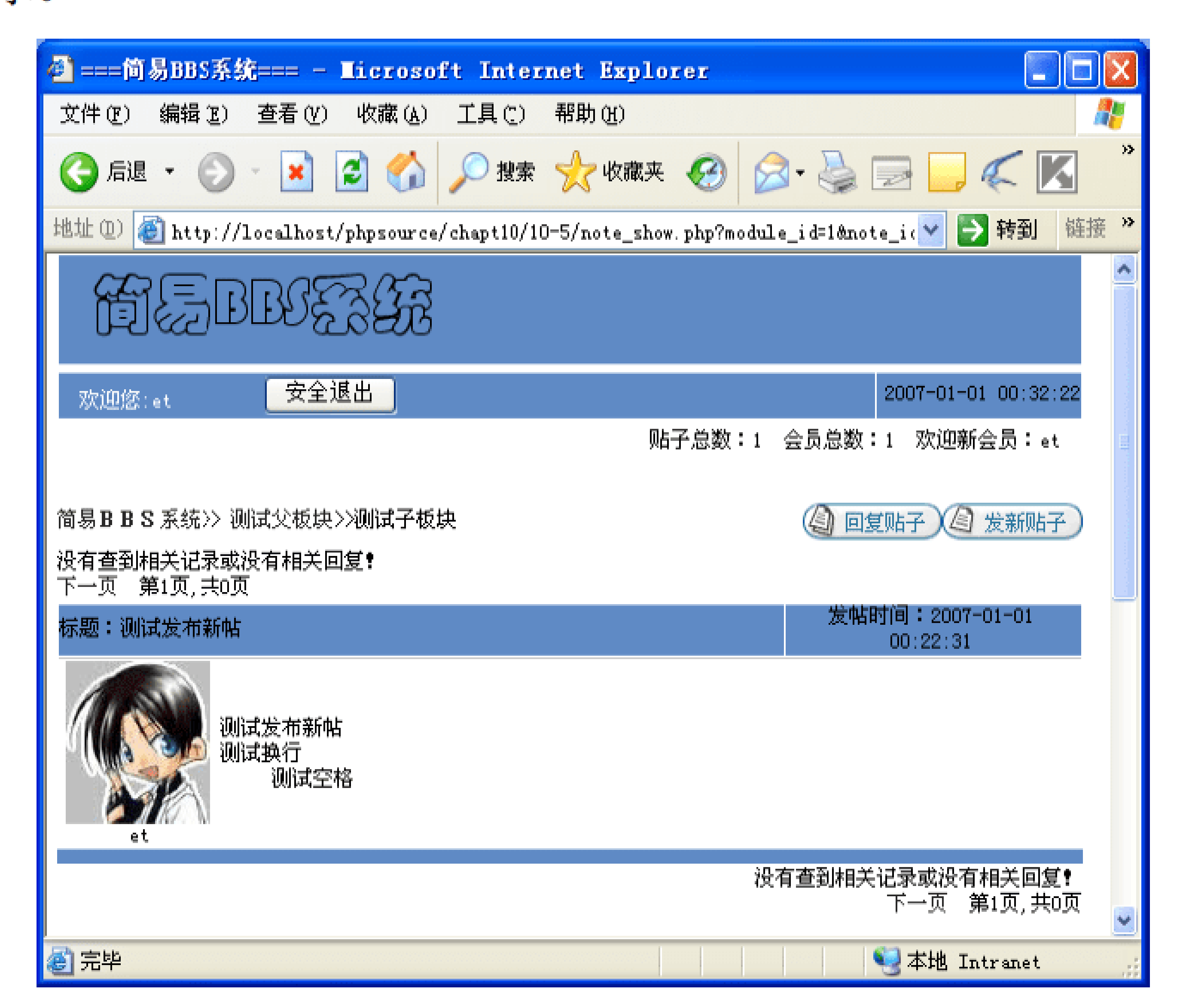


图 10-11 发新帖页面

（9）回复帖子。单击图 10-11 的“回复帖子”按钮，其运行效果如图 10-12 所示。

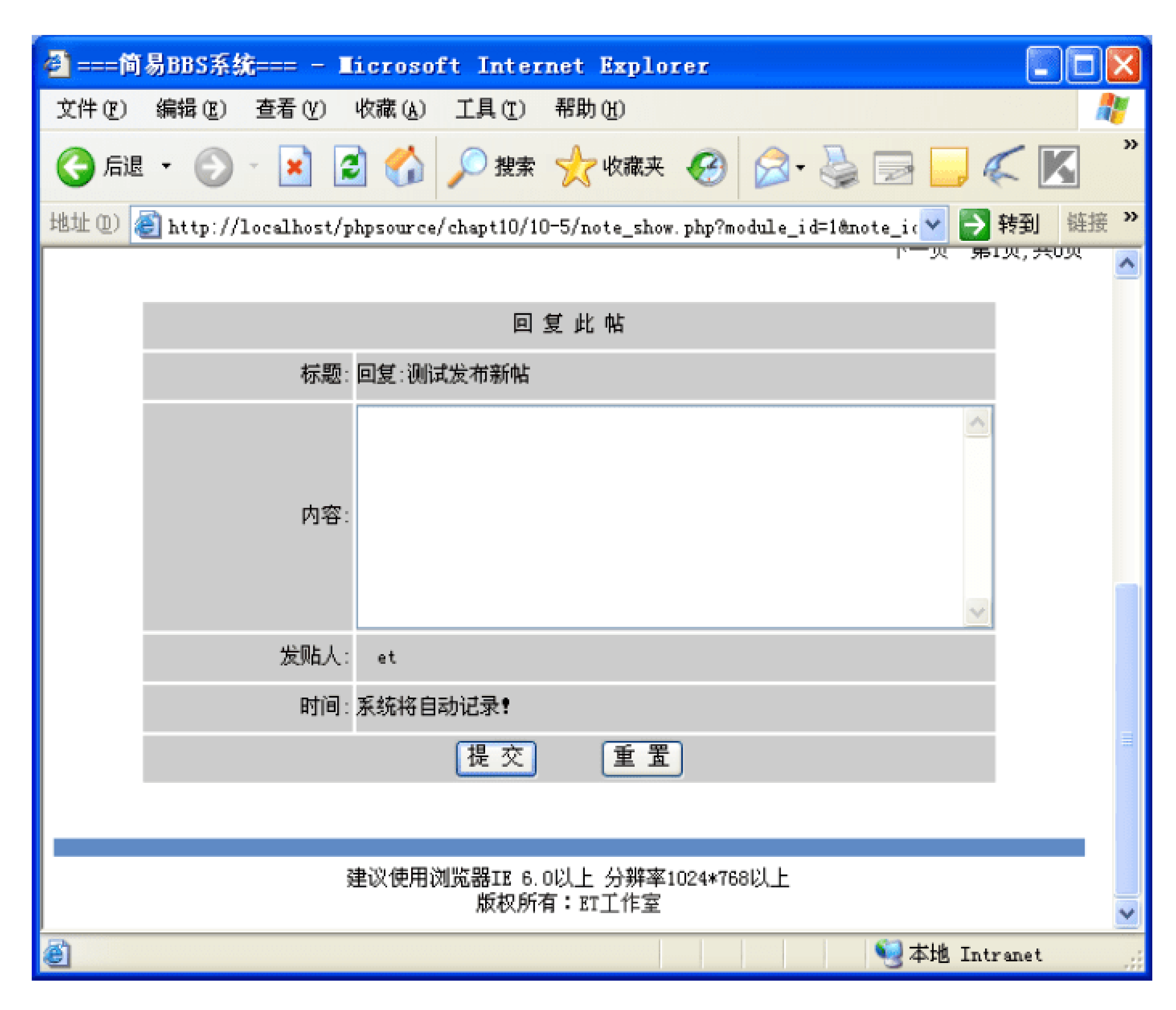


图 10-12　回复帖子页面

（10）再次查看帖子的具体内容。如步骤 8。效果如图 10-13 所示。



图 10-13　查看帖子页面

（11）进入管理界面的登录界面。在地址栏中键入 http://localhost/phpsource/chapt10/10-5/manage/，其运行效果如图 10-14 所示。

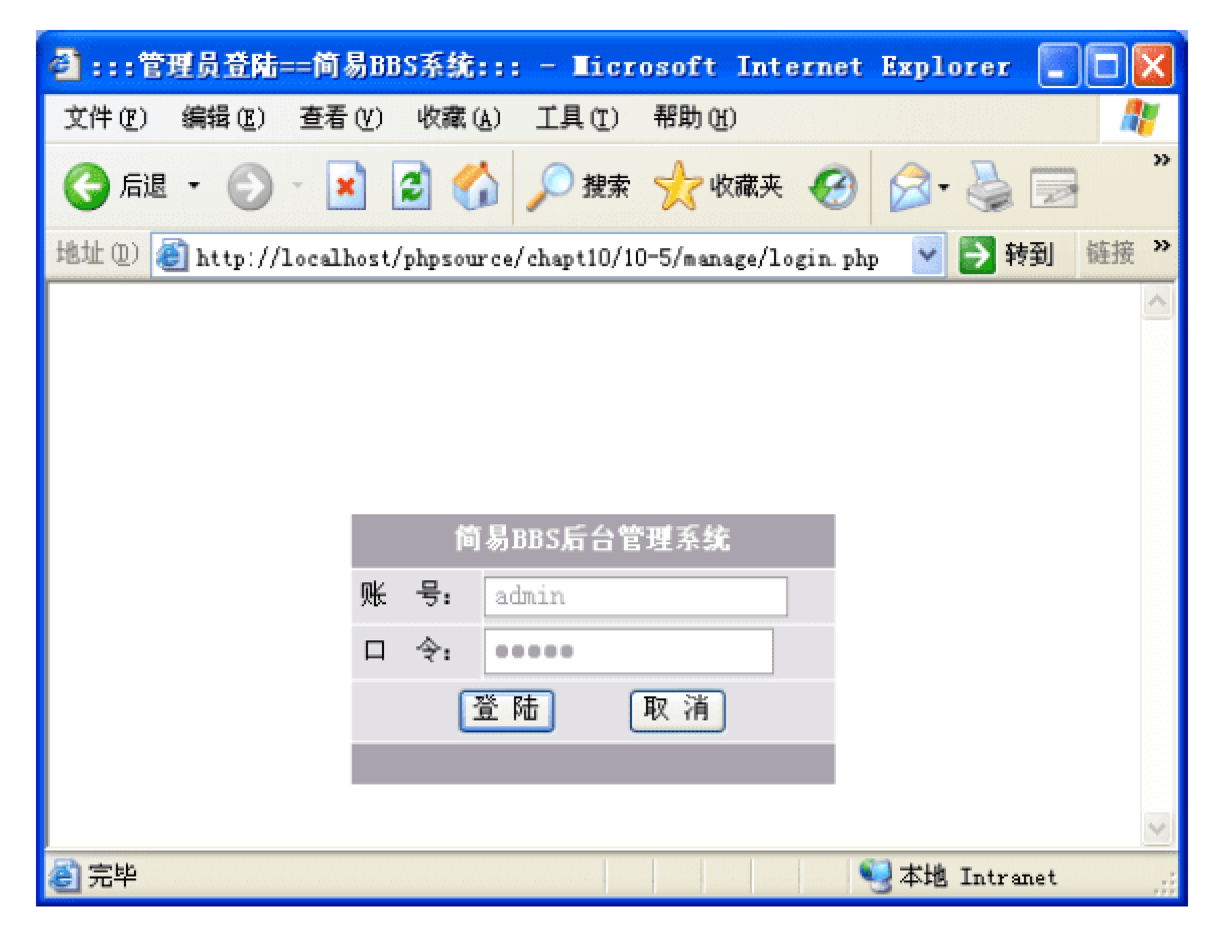


图 10-14　登录页面

（12）进入后台管理的主界面。账号、口令校验成功后，进入后台管理的主界面，效果图如图 10-15 所示。



图 10-15　后台管理主界面

其中各个管理菜单具体的界面如下：

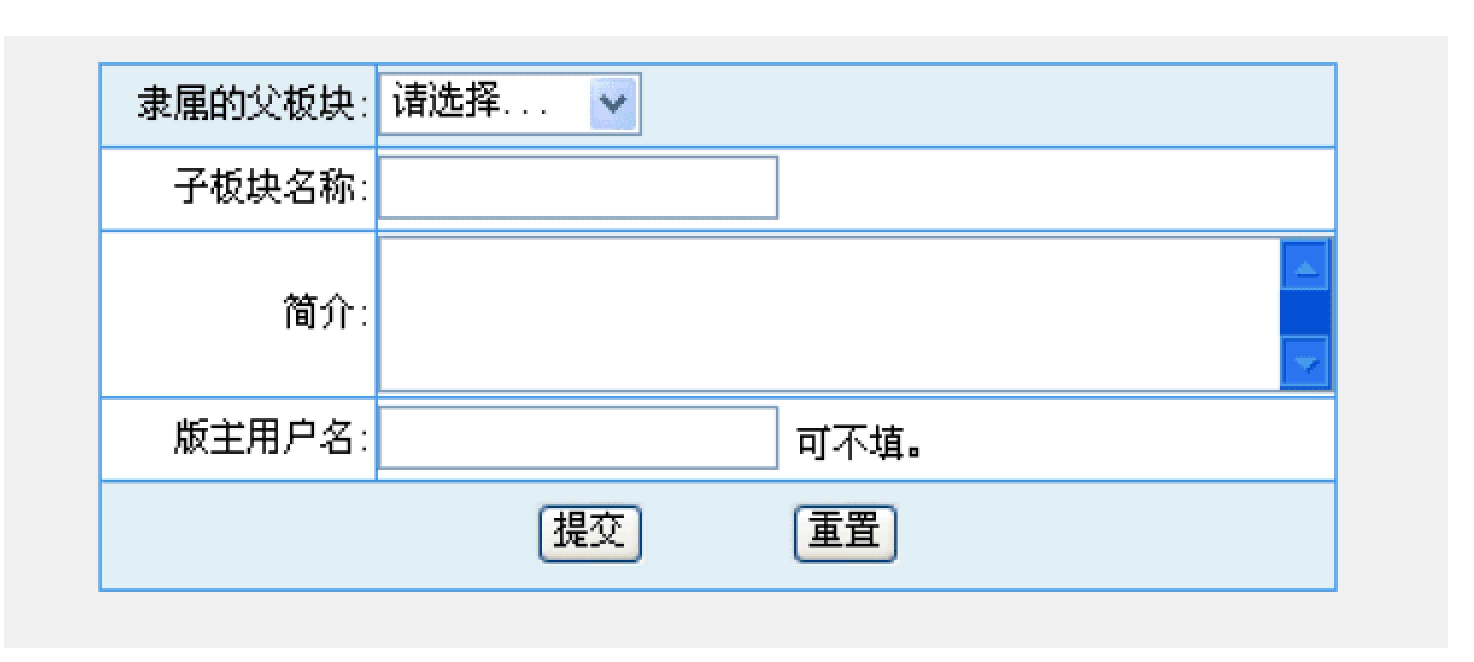
（1）父模块添加。

显示序号:	请填写一整数，如：1 。
父板块名称:	
提交　重置	

（2）父模块管理。

编号	显示序号	父板块名称	操作	
1	1	测试父板块	编辑	删除

（3）子模块添加。



（4）子模块管理。

显示序号	父板块名称	子板块名称	操作	
1	测试父板块			
	1	测试子板块	编辑	删除

（5）所有用户。

[1]下一页　第1页，共1页

序号	用户名	注册时间	最后登陆时间	操作
1	et	2007-01-01 00:14:50	2007-01-01 00:17:26	删除

[1]下一页　第1页，共1页

（6）用户检索。



（7）密码更改。

用户名：	admin
原口令：	
新密码：	
再次新密码：	
提 交　重 置	

（8）帖子管理。弹出新窗口，对任何帖子和任何回复均有删除权限。如图 10-16（a）和 10-16（b）所示。

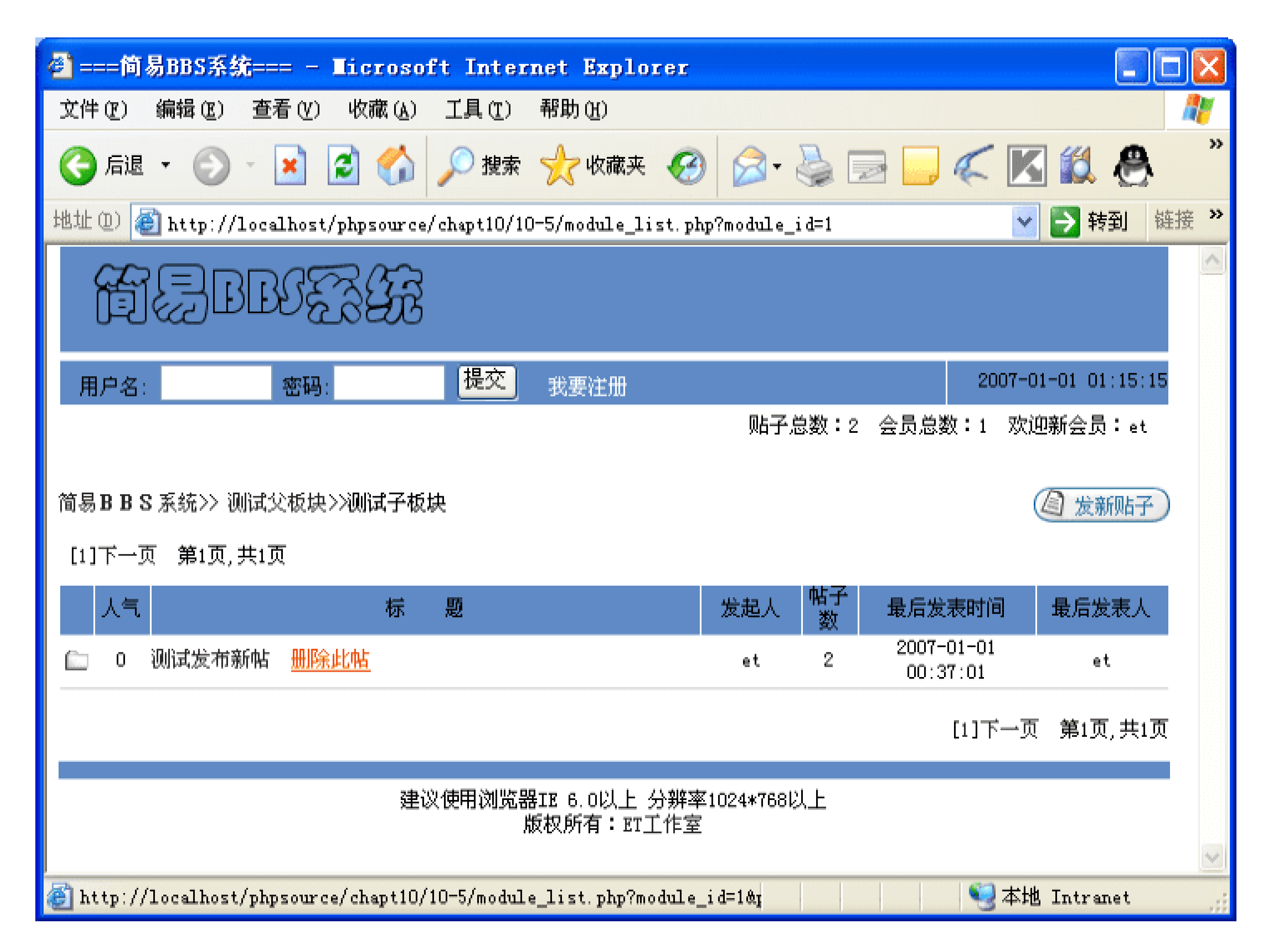


图 10-16（a） 帖子管理页面

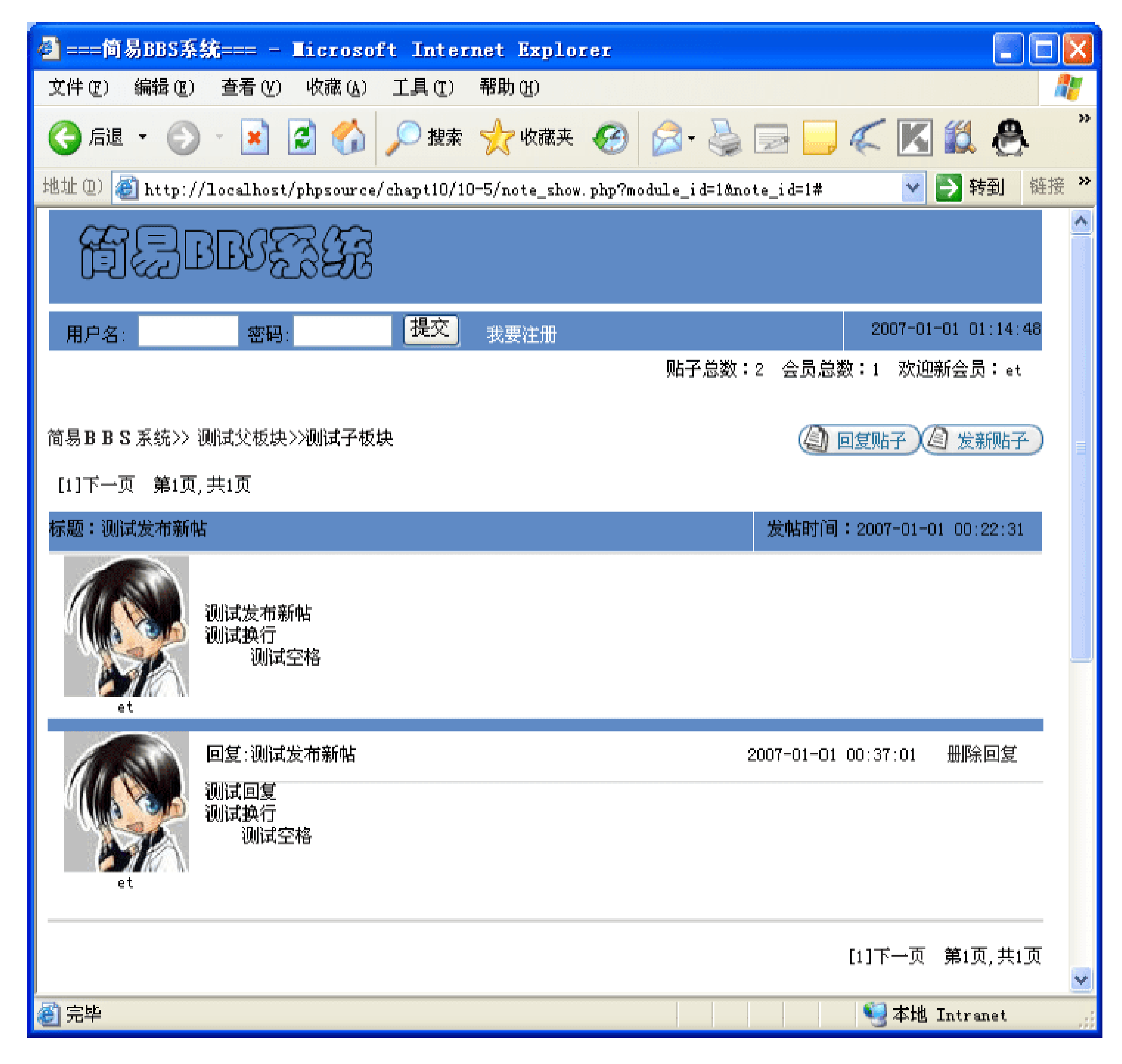


图 10-16（b） 帖子管理页面（展开后）

每个页面包含的二级页面或进行提交后的页面等就不再一一列举。

至此，一个简易 bbs 系统就开发完毕了。可以看出，即使是一个简易的论坛，也需要编制大量的程序。不过仔细分析不难发现，虽然程序很多，但是其操作不外乎数据库查询、插入、删除和修改等。这些基本操作的组合，可以设计出很多典型的应用程序。学习本程序的开发过程，除了学习一个简易论坛的开发思路外，还应该注意举一反三，通过本例的

开发过程，学会分析、设计其他 PHP+MySQL 的数据库应用程序。

10.6　本章小结

本章中通过一个简易 BBS 系统的开发实例，详细介绍了 PHP+MySQL 数据库开发的思路和方法。包括数据库设计的技巧、编程中的技巧、类和函数库的定义与使用、包含文件的使用等等，并且给出了全部核心代码。通过本章的学习，应当深入领会 PHP+MySQL 数据库编程的思路，掌握数据库编程中常见问题的解决方案，并举一反三，具备自行设计和开发其他应用系统的能力。

10.7　练习题

1. 在本章的简易论坛系统数据库设计中，父模块和子模块是通过什么建立关联的？
2. 在本章的简易论坛系统中，帖子主题和回复之间是通过什么建立关联的？
3. 如果实现论坛用户积分制度，即每发一篇帖子可以获得一定的积分，数据库应该如何修改？程序应当如何编写？并请试实现之。
4. 为开发好的程序建立安装程序有何好处？